

UiO : **Department of Informatics**  
University of Oslo

# Verbal Inference

A Study of Semantic Entailment

Ole Johan Ljosland  
Master's Thesis Spring 2014





# Verbal Inference

Ole Johan Ljosland

6th May 2014



# Abstract

In the field of lexical semantics, there have been many suggestions as to how the meaning of sentences and utterances should be represented. Such theories typically make use of traditional logic frameworks, and the various words that appear in sentences, acting as subjects or objects, for instance, are mapped from the syntax to proposed semantic structures. When such structures share common rules and an overall framework, they can be related to each other in various ways. This can then be used to capture an important concept in both traditional logic, and in semantics; the notion of inference, a process that involves arriving at new truth conclusions based on given truth premises. This thesis will focus on the meaning of verbs and the entailment relations that might occur between them. It will be looking at three specific ways of representing the semantics of verbal situations, and how they capture specific entailment phenomena. The idea will be to compare the three frameworks, and to arrive at what each of them does best when it comes to verbal inference.



# Contents

<b>1</b>	<b>The Goal of this Thesis</b>	<b>1</b>
<b>2</b>	<b>Basic Linguistic Knowledge</b>	<b>3</b>
2.1	Lexemes and Senses . . . . .	3
2.1.1	Relations between Senses . . . . .	4
2.2	Thematic Roles . . . . .	5
2.3	Inference and Entailment . . . . .	7
<b>3</b>	<b>Jackendoff's Conceptual Structures</b>	<b>11</b>
3.1	Overview . . . . .	11
3.2	Semantic Concepts and Semantic Functions . . . . .	13
3.2.1	Basic Functions . . . . .	14
3.2.2	Aspectual Functions . . . . .	15
3.2.3	Causative Functions . . . . .	16
3.2.4	Other Functions that Map to Semantic Concepts . . .	17
3.2.5	Modifying Functions . . . . .	17
3.3	Semantic Fields . . . . .	18
3.4	Semantic Features . . . . .	19
3.5	Combinatory Functions . . . . .	21
3.5.1	Including Functions . . . . .	21
3.5.2	Excluding Functions . . . . .	22
3.6	Thematic Roles in Conceptual Structures . . . . .	22
3.7	Inference Rules . . . . .	23
3.8	Flat Conceptual Structures . . . . .	24
<b>4</b>	<b>Parsons' Underlying Event Theory</b>	<b>27</b>
4.1	Overview . . . . .	27
4.2	Thematic Roles . . . . .	29
4.3	Modifier Predicates . . . . .	30
4.4	Representing Time . . . . .	32
4.5	Causatives and Inchoatives . . . . .	33
4.6	Progressive and Perfect Aspects . . . . .	35
4.7	Inference Rules . . . . .	36
<b>5</b>	<b>VerbNet: A Lexical Database</b>	<b>39</b>
5.1	Overview . . . . .	39
5.2	Verb Alternations and Verb Classes . . . . .	39
5.3	The Structure of VerbNet . . . . .	41

5.3.1	A Verb Class Example . . . . .	42
5.3.2	The Class Hierarchy . . . . .	43
5.3.3	VerbNet's Semantic Structures . . . . .	44
5.3.4	VerbNet's Approach to Inference . . . . .	46
5.4	Other Lexical Resource Databases . . . . .	48
5.4.1	WordNet . . . . .	48
5.4.2	PropBank . . . . .	50
5.4.3	FrameNet . . . . .	51
<b>6</b>	<b>Planning the Project</b>	<b>53</b>
<b>7</b>	<b>Verbs of Contact by Impact</b>	<b>55</b>
7.1	Stab . . . . .	55
7.1.1	Jackendoff's Conceptual Structure Representation . .	56
7.1.2	Parsons' Underlying Event Representation . . . . .	58
7.1.3	VerbNet's Semantic Representation . . . . .	59
7.1.4	Notable Differences and Similarities . . . . .	62
7.2	Hit . . . . .	63
7.2.1	Jackendoff's Conceptual Structure Representation . .	64
7.2.2	Parsons' Underlying Event Representation . . . . .	66
7.2.3	VerbNet's Semantic Representation . . . . .	67
7.2.4	Notable Differences and Similarities . . . . .	69
<b>8</b>	<b>Verbs of Killing</b>	<b>71</b>
8.1	Kill/Die . . . . .	71
8.1.1	Jackendoff's Conceptual Structure Representation . .	72
8.1.2	Parsons' Underlying Event Representation . . . . .	72
8.1.3	VerbNet's Semantic Representation . . . . .	73
8.1.4	Notable Differences and Similarities . . . . .	74
<b>9</b>	<b>Verbs of Putting: <i>Spray/Load</i> Verbs</b>	<b>77</b>
9.1	Load . . . . .	77
9.1.1	Jackendoff's Conceptual Structure Representation . .	78
9.1.2	Parsons' Underlying Event Representation . . . . .	79
9.1.3	VerbNet's Semantic Representation . . . . .	81
9.1.4	Notable Differences and Similarities . . . . .	82
<b>10</b>	<b>Verbs of Ingesting</b>	<b>85</b>
10.1	Eat . . . . .	85
10.1.1	Jackendoff's Conceptual Structure Representation . .	85
10.1.2	Parsons' Underlying Event Representation . . . . .	86
10.1.3	VerbNet's Semantic Representation . . . . .	87
10.1.4	Notable Differences and Similarities . . . . .	87
<b>11</b>	<b>Verbs of Motion</b>	<b>89</b>
11.1	Jump . . . . .	89
11.1.1	Jackendoff's Conceptual Structure Representation . .	90
11.1.2	Parsons' Underlying Event Representation . . . . .	92
11.1.3	VerbNet's Semantic Representation . . . . .	93



11.1.4	Notable Differences and Similarities . . . . .	94
11.2	Climb . . . . .	96
11.2.1	Jackendoff's Conceptual Structure Representation . .	97
11.2.2	Parsons' Underlying Event Representation . . . . .	98
11.2.3	VerbNet's Semantic Representation . . . . .	99
11.2.4	Notable Differences and Similarities . . . . .	101
<b>12</b>	<b>Verbs of Change of Possession</b>	<b>105</b>
12.1	Buy/Sell/Pay . . . . .	105
12.1.1	Jackendoff's Conceptual Structure Representation . .	106
12.1.2	Parsons' Underlying Event Representation . . . . .	108
12.1.3	VerbNet's Semantic Representation . . . . .	109
12.1.4	Notable Differences and Similarities . . . . .	111
<b>13</b>	<b>Verbs of Change of State</b>	<b>115</b>
13.1	Open . . . . .	115
13.1.1	Jackendoff's Conceptual Structure Representation . .	116
13.1.2	Parsons' Underlying Event Representation . . . . .	117
13.1.3	VerbNet's Semantic Representation . . . . .	118
13.1.4	Notable Differences and Similarities . . . . .	119
<b>14</b>	<b>Strengths and Weaknesses</b>	<b>121</b>
14.1	Coverage . . . . .	121
14.2	The Mapping from Syntax to Semantics . . . . .	122
14.3	Generality versus Specificity . . . . .	123
14.4	Structure Formats . . . . .	124
14.5	Thematic Role Labeling . . . . .	125
14.6	Semantic Structure Detail . . . . .	127
14.7	Tense and Aspect, and the Representation of Nouns . . . . .	128
<b>15</b>	<b>Conclusion</b>	<b>129</b>



# Preface

This work is a master's degree thesis for the Informatics: Language and Communication program at the Department of Informatics (IFI), situated under The Faculty of Mathematics and Natural Sciences at the University of Oslo (UiO). It is an original and independent product created by the author, Ole Johan Ljosland. The master's program was started in August 2012, which was the author's 5th semester at IFI, and ended in May 2014, the author's 8th semester.

The author wishes to thank his supervisor, Jan Tore Lønning, professor at the Research Group for Language Technology, for offering excellent guidance and motivation through the process of planning and writing this thesis.



# Chapter 1

## The Goal of this Thesis

The name of this thesis is *Verbal Inference*, so the focus will be on the relationship between sentences that make use of specific verbs and different arguments to these verbs. The concept of verbal inference involves examining given sentences, and establishing what other sentences are also true, based on the meaning behind the initial sentences. In other words, a sentence describing a situation might entail, or logically lead to, other sentences describing similar situations.

Although there are many different ways of representing semantics, as mentioned, this thesis will focus on three theories in particular, and how each deals with the notion of verbal inference. Naturally, these three ways of representation all focus on verbal situations, and how verbs can be given specific logic structures, or *lexical entries*. The three theories are the conceptual structure representation proposed by Ray Jackendoff, the underlying event formulas employed by Terence Parsons, and the semantic structures provided by the VerbNet lexical database, which bases most of its structure on the verb classes given by Beth Levin.

The main goal of this thesis will be to go through various entailment phenomena, or situations involving different sentences which may be related to each other by means of entailment. Each example will examine how the three semantic theories structure and cope with the given entailment phenomenon. In the end, a conclusive overview will be given, which will summarize the various differences and similarities between the three representations, as well as what their strengths and weaknesses are with regards to verbal inference. The idea is to arrive at which of the theories are the best at capturing specific entailments.

Before examining the entailment examples, these first chapters will provide a basic overview of necessary linguistic terms and concepts, as well as a quite thorough introduction for each of the three semantic representations. So, chapter 2 will provide the basics needed, including a more formal definition of the important terms *inference* and *entailment*, while chapters 3, 4 and 5 will introduce the semantic structures provided by Jackendoff, Parsons and VerbNet, respectively. Chapter 6 will explain the thesis' planning process, while chapters 7 through 13 will examine specific entailment examples. Chapter 14 will offer a conclusive overview

of the strengths and weaknesses of each framework. Finally, chapter 15 will provide an illustrative table and a final conclusion.

## Chapter 2

# Basic Linguistic Knowledge

As mentioned, this chapter will present some important terms and concepts that will be used quite frequently in this thesis. These are, in other words, necessary to understand in order to fully apprehend the different semantic representations that will be presented in the coming chapters. The concepts include, most importantly, the actual definition of a word (specifically verbs) in section 2.1, the various ways these are related to each other (section 2.1.1), the arguments, or participants that occupy a situation (section 2.2), and the definition of verbal inference and semantic entailment (section 2.3).

### 2.1 Lexemes and Senses

When modelling the semantics of words, it's important to know that the idea of a word can actually have many different definitions. In a lexicon, a word is typically listed with a proposed definition, and this representation is traditionally called a *lexeme*. The graphical form often used to represent a lexeme is called a *lemma*, which acts as the headword of the given lexeme. Furthermore, mapping from a word form to the lemma is called *lemmatization*. For example, the verb lemma *read* would represent all word forms possible under that lemma, like *reads*, *reading* etc., and lemmatizing those forms would yield the same lemma [16, p. 645].

A word by itself may mean many different things. These meanings can be called *senses* and they represent the concepts a word realizes given the context. For example, the verb *drill* might be associated with the activity of making a hole using a tool, or to the activity involving a strict form of training. The different senses a word can have are traditionally listed under a chosen lemma entry in a lexicon, making up a lexeme [16, p. 646]. The example in (1), for instance, shows how the two possible verb senses of *drill* could be listed in a lexicon [12, p. 467]. Note that the whole structure is the actual lexeme and the name *drill* is the lemma used to represent the lexeme.

- (1) *drill*:
  - a. Verb<sub>1</sub>: *to make a hole in something, using a drill.*

- b. Verb<sub>2</sub>: *to teach somebody to do something by making them repeat it a lot of times.*

### 2.1.1 Relations between Senses

Different senses are related to one another in a variety of ways. When a lemma can have different senses that are totally, or almost totally, unrelated then that lemma is typically listed several times in the lexicon; one representation for each of the unrelated senses. Such unrelated senses are called *homonyms*. The word *bank*, for example, traditionally has at least two lexemes in a lexicon, one for when the word is connected to the notion of banks along a river, for instance, and one for when the meaning is connected to a kind of building. It must be said, though, that deciding whether senses are homonyms may be very hard, since the distinctions between them might not be that easily defined. Because of this, different dictionaries may choose to represent senses in distinctive ways, which may differ among them.

The two related terms *homophones* and *homographs* both denote relations among lemmas instead of senses. Homophones are pronounced the same way but written differently, as in *two* and *too*, while homographs are pronounced differently, but written the same, as in the two pronunciations of the word *bass*. These similarities have typically come about coincidentally, as it just so happens that the same word was used to describe two different meanings. This idea also applies to homonyms.

The senses listed in a single lexeme are semantically related, so the term *polysemy* is used instead of homonymy. As an illustration, take the sentences in (2) and (3) below.

- (2) *John deposited money to the bank.*  
(3) *John gave blood to the bank.*

Here, *the bank* has two different senses, meaning a bank that stores either money or blood, respectively. But they are still semantically related, both denoting a building that stores something. Because of this, the two senses are listed in the same lexeme, meaning that there is a polysemy relation between them.

A subtype of polysemy, called *metonymy*, occurs when one aspect of a concept refers to another aspect of the same concept. This aspect may also represent the whole concept. It is possible, for example, to say sentence (4) instead of sentence (5). In sentence (4), *Stephen King* represents the concept of *the books written by Stephen King*, which is the concept in sentence (5).

- (4) *He loves reading Stephen King.*  
(5) *He loves reading books written by Stephen King.*

The term *synonymy* is used for the relation between senses that represent the exact (or close to exact) same meaning. Typical examples are the word pairs *couch* and *sofa*, and *car* and *automobile*. However, there will probably never exist completely true synonyms, where the two



words stand for the exact same meaning in all contexts. This is because there is always some small difference between the given senses, however insignificant, and in the way each can operate according to the given context.

Related to synonymy is the term *antonymy* which deals with the opposite meanings of senses. There are three specific kinds of antonymy. It can be completely *discrete*, as in the relation between the words *dead* and *alive* or *on* and *off* etc., where there is no middle ground between the two meanings. It's not possible, for instance, to say that something is almost off, or almost alive. When there is a *middle ground*, on the other hand, as in the relationship between the adjectives *hot* and *cold*, *small* and *large* and so forth, they are defined according to a gradable scale. You could, for instance, say that something can become hotter or colder. Finally, antonyms may also be *reversives* where there is movement in either one direction or the other. Examples include *up* and *down*, and *rise* and *fall*.

Another relation that is useful, which only applies to nouns, is *hyponymy*, which involves senses that have subclasses, or subsenses. These act as further specifications where all properties of a given sense is inherited to some other, lower sense. The noun *dog*, for instance, can be a subsense of *animal* in that it specifies what kind of animal it is. *Dog* is then a *hyponym* of *animal*, and *animal* is the *hypernym*, or superordinate sense, of *dog*. Related senses and subsenses may be organized in a *taxonomy*, a tree-like structure that can show the hyponymy relations among them.

A relation called *troponymy* is basically the same as hyponymy, except that it applies to verbs. It denotes a relation in which a verb sense is said to specify in which manner the action of another, related verb sense is done. For example, *sprinting* is a certain type of *running*. So, *sprint* is, in other words, a troponym of *run* [18].

Another noun relation called *meronymy*, not to be confused with metonymy, is the term used when senses are part of other senses. This relation is also called the part-whole relation. For example, *finger* is a *meronym* of *hand* because it is a part of it. *Hand*, in turn, is the *holonym* of *finger* in this case [16, p. 646-651].

Finally, the *entailment* relation may also occur between verb senses where one sense is said to lead to or infer another sense. In other words, if one sense of a word is true then it might also be that some other sense is also true. For example, when a person snores it is logically entailed that the person is also sleeping, hence *snore* entails *sleep* [19, p. 40]. This kind of relation is very much related to *semantic entailment*, which is of course the main topic of this thesis. As mentioned, section 2.3 will explain this important concept, and the notion of verbal inference, in more detail.

## 2.2 Thematic Roles

A category of particular importance is the notion of thematic roles, a concept that gained particular relevance ever since Jeffrey Gruber [11] and Charles J. Fillmore [10] began focusing on it. When you look at different

verbs and their arguments it is easy to see that they share many similarities. For example, in sentences (6) and (7) below, it is apparent that Mary acts as both an "opener" and a "closer"; she *opens* the door in sentence (6), but *closes* it in sentence (7).

(6) *Mary opened the door.*

(7) *Mary closed the door.*

These are very specific roles, different for each verb. When roles are this specified they are sometimes also called *deep roles* [16, p. 654]. But they are, in fact, very similar, because they both denote someone acting deliberately in causing a situation (opening/closing the door). This more general role is typically called an *Agent* or *Actor*, and this is but one example in the large list of thematic roles that have been proposed in the linguistic literature. Other examples of roles are *Experiencer*, where an entity does not have control over what is happening, *Theme*, which acts as the thing directly affected by an action, and *Instrument*, which is the thing used in the action [16, p. 654-655] [26, p. 153-155] [3, p. 472]. So, in other words, all situations, which are represented using sentences, describe a number of participants acting in various ways according to one another.

Most theories address the situations denoted by sentences as *events* [16, p. 597], though some also use the term *eventualities* [25, p. 20-21], which covers both active situations where something happens (events) and *states*, which are just static representations of certain circumstances. Note that eventualities can be further divided into the types *accomplishment*, *achievement* and *process*, though these will be explained in more detail in section 4.4.

Verbs may denote any eventuality (an event or a state), and the participants in such eventualities are labeled using the thematic roles. The theories that will be presented deal with events and states in different ways, so their exact definitions and use will be explained in more detail in the coming chapters. For now, though, the sentences in (8) show two examples of eventualities, where sentence (8a) is a state and sentence (8b) is an event.

(8) a. *The earth is round.*

b. *He throws the ball.*

The three approaches to semantic representation also deal with participants slightly differently, though they all use thematic roles to refer to them. As such, each theory uses different rules for defining the thematic roles and for the specifics involved linking them with the arguments in the syntax. The lists of thematic roles therefore vary a bit among them, though the most important ones, such as those mentioned above, are all shared. All three would, for instance, say that both *the earth* and *the ball*, in sentence (8), at least have the thematic role of Theme.

In spite of these differences, some universal rules apply when dealing with thematic roles. All verb arguments, that is the syntactic constituents that operate according to the verb (such as the subject and object), are typically given a unique thematic role. So, there are, for instance, typically

no more than one Agent in an event; if one participant occupies a role then no other participant can also have that role. Furthermore, participants are typically never given more than one role label. In other words, a participant is never both a Theme and an Agent, for instance. However, in the coming chapters we will see some examples where these rules are slightly altered, particularly in Jackendoff's and Parsons' theories (chapters 3 and 4).

One interesting alternative method to a listing of various thematic roles is to use even more generalized semantic roles. Here, the roles only express their rough meaning, each having some heuristic features that define the role. These generalized roles cover sets of more specific roles. For example, a generalized Agent, sometimes called a *Proto-Agent* [8], would represent all roles that have Agent-like qualities. The likelihood that an argument is labeled with a generalized role depends on which features the argument exhibits. Consider, for instance, the sentence in (9).

(9) *David threw the ball.*

Here, *David* is the cause of the ball being thrown and he is also doing it intentionally, all features that are typical of a Proto-Agent. Since no other arguments have more Proto-Agent features in the example, *David* is labeled as the Proto-Agent [16, p. 657]. This alternative is not used by any of the semantic theories that will be exploring, however.

## 2.3 Inference and Entailment

The term *inference* involves arriving at or deducing other facts based on some already established information. Basically, given truths may lead to other truths, if they are logically entailed, and inference is then the act of finding such truths. In other words, Jurafsky and Martin [16, p. 585] define inference as the "ability to draw valid conclusions based on the meaning representation of inputs and [the system's] store of background knowledge."

Furthermore, *entailment*, or logical consequence, is the term used to denote a relationship between structures of meaning, which can be anything from sentences to logical formulas. Such relations say that statements may lead to, or follow, other statements, because they share the same truths [2]. So, if entailment means that there exists paths from one set of truths to other truths, then inference is the procedure used to travel those paths. Note that the entailment relation between words, mentioned in section 2.1.1, is just a specific kind of the overall concept of entailment.

Sentences that make use of verbs in accordance with both obligatory and non-obligatory arguments, may entail, or lead to, other sentences which use the same verb or a different verb altogether. *Verbal* inference then involves deducing the bridge between such related sentences. Furthermore, *semantic* entailment is the name of the actual entailment relation that happens between the meanings of different sentences, or utterances [3, p. 18-23]. To give another definition, semantic entailment is said to be "the problem of determining if the meaning of a given sentence entails that

of another" [27, p. 1]. To give some introductory examples, consider the sentences in (10), (11) and (12).

- (10) a. *Jack threw a ball.*  
b. *Jack threw something.*
- (11) a. *Mary ate a sandwich.*  
b. *Mary ate.*
- (12) a. *Brutus killed Caesar.*  
b. *Caesar died.*

From sentence (10a) we know that Jack threw a specific thing, namely a ball, but in sentence (10b) we only know that he threw something, not specifically what it was. The fact that Jack threw something is true in both sentence (10a) and sentence (10b); if Jack threw a ball then it is also true that Jack threw something. However, the fact that Jack threw a ball is only a fact in sentence (10a), not in sentence (10b), as the latter might involve some other object having being thrown. Because of this, we say that sentence (10a) implies, or entails, sentence (10b), but not vice versa. However, because this entailment example concerns the relationship between nouns, not verbs or different verbal constructions, the issue will not be explored in detail in this thesis. It's mentioned here only because it serves as an notable example of semantic entailment.

In a similar fashion to the sentences in (10), sentence (11a) entails sentence (11b), even though one argument is missing in the latter (it is intransitive, whereas sentence (11a) is transitive). The sentence in (11b), on the other hand, does not entail sentence (11a), which is also similar to the sentence relations in (10). In other words, if Mary ate a sandwich it is true that Mary also ate, but if Mary only ate then it might not be true that the eating involved a sandwich.

The sentence in 12a) entails sentence (12b), but not vice versa, because if Caesar was killed then he also died, but if Caesar died then it might not be the case that he was killed (he might have died from some other cause). This is an example of inference happening between verbs that do not share the same lemma (the opposite case was true for the examples in (10) and (11)). As seen, the word *entail* is what will be used the most when describing inference instances, where one eventuality leads to (or entails) one or multiple eventualities.

Note that, in the above examples, the sentence pairs all describe events that take place at the same time. In example (10), for instance, this was an event of throwing, only it was referred to slightly differently in the two given sentences. If we say that sentence (10a) entails sentence (10b), then they are basically the same event, taking place at the same given time. There might, however, also be sentences that entail events that happened at some other time. An example of this can be seen in example (13), where there are two separate events; an event of marrying and an event of divorcing.

- (13) a. *John married Carrie.*  
b. *John divorced Carrie.*

These are not happening at the same instant, but rather at two separate points in time. The sentence in (13b) should entail sentence (13a) because one cannot divorce someone unless you already married that someone. In other words, the event in sentence (13b) can't happen unless the event in sentence (13a) also happened prior to it, so there is an entailment relation between them. Some of the entailment examples that will be explored will examine verbs that may refer to different kinds of events, though are still related in meaning.

So, the problem that semanticists face is to establish a solid logical framework that represents semantic meaning, which also captures all kinds of verbal entailment phenomena. The next chapter will present Jackendoff's suggested system, while the following two chapters will each look at Parsons' and VerbNet's proposed frameworks, respectively.



## Chapter 3

# Jackendoff's Conceptual Structures

### 3.1 Overview

Conceptual structure [...] is a part of thought. It is the locus for the understanding of linguistic utterances in context, incorporating pragmatic considerations and "world knowledge"; it is cognitive structure in terms of which reasoning and planning take place."

— Ray Jackendoff, [13, p. 123]

Ray Jackendoff's theory of conceptual semantics is a compositional theory of meaning, aiming to provide a linking between syntax and semantics. It employs simple primitive components describing fundamental semantic concepts, or mental representations, which users of a language can understand when they are expressing them. The meaning of a word or a sentence can, as the theory emphasizes, be represented as a conceptual structure, ranging from being just a simple primitive to being a complex conjunct involving the embedding of several semantic components. The whole meaning of the structure then depends on the individual meaning of each component.

When discussing Jackendoff, the focus will be on how his theory is presented in Jackendoff [15], because it is the most all-encompassing, presenting the clearest version of the theory, and the biggest number of definitions of specific verbs available in a single work. However, some of the additions in later revisions (such as Jackendoff [13]) will also be presented.

The conceptual structures are used in order to describe inferences such as the ones occurring with sentences (14), where if the case is true that George killed a dragon, then that dragon ended up dying. This inference can be paraphrased as the schema shown in sentence (14c), where X and Y can be substituted for any entity [26, p. 278-279].

- (14) a. *George killed the dragon.*  
b. *The dragon died.*

- c. *X killed Y entails Y died.*

In other words, if *X killed Y* then it is certain (entailed) that *Y died*. This particular pattern can also be applied to a multitude of other verb pairs, such as *lift* and *rise*, *give* and *receive* and *persuade* and *believe*, shown in (15).

- (15) a. *X lifted Y entails Y rose.*
- b. *X gave Z to Y entails Y received Z.*
- c. *X persuaded Y that P entails Y came to believe P* (where *P* is a proposal).

There is an apparent similarity between these schemas, as they all involve *X* doing something that leads to something happening to *Y*. In order to capture this, Jackendoff proposes the more general schema shown in (16).

- (16) *X causes E to occur entails E occurs* (where *E* is an event).

In other words, there is, as can be seen by the symbol *E*, a common semantic element or concept known as an *event*, which should be part of a theory of semantic representation. Another semantic element that is present in this schema is the term *cause*, and that component should also be encoded into a semantic structural theory.

The two common elements event and cause are only examples of the many primitive semantic components that lies underneath the meaning of sentences, in the so-called deep structures [4]. Jackendoff's conceptual theory aims to provide a list of such universal elements, each categorized as a specific type. Here, *event* was an example of what is referred to as a universal category, or semantic concept, while *cause* is instead a semantic function, taking semantic concepts as arguments and producing another semantic concept. These different components, plus two others, will be explained in more detail in section 3.2, but first, the actual formation of a conceptual structure, i. e. the specific notation used by Jackendoff, will be introduced.

The simplest conceptual structure is just a single semantic concept representing some mental representation, let's say a party. The standard notation involves using enclosed square brackets to indicate a semantic concept and labeling it with the specific semantic concept type, in this case an event. The final structure is shown in (17). Notice that the semantic concept type is in lower case letters, while the semantic function giving the concept is in all uppercase letters.

- (17) [event PARTY]

In other words, *PARTY* is a semantic function that takes no arguments and returns an event, a semantic concept. So in this case, *PARTY* acts just as the name of a specific type of event, without having any inner structure. Rather than complex, it is instead an unanalyzed atomic component. A semantic concept might also not even have an inner function, being just an



empty structure, but with the concept label still showing. An example is *[thing ]*, which constitutes just an unspecified thing.

Next, let's look at a slightly more complex example, involving the multiple argument-taking function *CAUSE*. When semantic functions take arguments, these are enclosed inside parentheses and separated by commas. *CAUSE* takes two arguments; the causer, which can be a thing or an event, and the effect, which must be an event. So, for example, the conceptual structure for the (slightly constructed) sentence shown in (18a) would have the structure shown in (18b) [15, p. 43]. Notice that the arguments to *CAUSE* are also in themselves semantic concepts, so they are also enclosed in square brackets.

- (18) a. *Kim caused a party to happen.*  
b. [event CAUSE ([thing KIM], [event PARTY])]

So, the whole meaning can be summarized as such: there is a causing event which has an inner structure involving the causer *Kim*, which is a thing, and the effect *party*, which is an event.

As we have seen, the semantic structures that Jackendoff employs consist of semantic concepts (universal categories) and semantic functions, where the latter maps between the former. The semantic functions might also have certain semantic features applied, which are often specific for each function. *CAUSE*, for example, might have a feature involving success; whether the event successfully completes or not. Another important notion is the nature of what semantic field is in effect, something which constitutes the applied meaning of a structure, and might also specify which types the semantic functions can take as arguments. In the following sections, these four theoretic components, namely semantic concepts, functions, fields and features, will be explained in more detail. Also, section 3.8 will introduce an alternative form of the structures provided by Jackendoff's theory.

### 3.2 Semantic Concepts and Semantic Functions

In addition to the semantic concepts already mentioned, namely *event* and *thing*, Jackendoff operates with a few others as well. Some of these include *states*, *paths*, *places*, *properties*, *manners* and *times*, where events and states are the most basic concept types [15, p. 43] [26, p. 279]. All of them can be combined in various ways to yield a large number of different conceptual structures. As such, a single complex structure might contain a number of concepts which have a different type than the overall structure. When it comes to which syntactic constituents can represent which semantic concepts, it seems noun phrases can express almost any of them, while prepositional phrases can express both places, paths and properties. Furthermore, a whole sentence may express either an event or a state.

As mentioned, the semantic functions map between semantic concepts. Section 3.2.1 will go through the most important of them, which Jackendoff

categorizes into three groups. These are basic functions, aspectual functions and causative functions.

Each function will have the pattern  $\langle FUNCTION \rangle (argument-01, argument-02, \dots argument-N) \rightarrow \langle Semantic\ Concept \rangle$ , where N is the number of arguments the function takes,  $\langle FUNCTION \rangle$  is the name of the function in question, and  $\langle Semantic\ Concept \rangle$  is the name of the concept the function maps to. If an argument can be of different types then those types are shown together, divided by the / symbol. The functions are shown with an explanation in the *b.* sections, and an example in the *c.* sections. The examples first show the surface sentence, followed by the conceptual structure (after the  $\rightarrow$  symbol).

### 3.2.1 Basic Functions

The following basic functions either map to states, events or paths, and include *BE*, *STAY*, *GO*, *EXT*, *ORIENT*, *MOVE*, *CONF*, *FROM* and *TO* [15, p. 43].

- (19) a. *BE* (thing, place)  $\rightarrow$  state  
b. A thing is in a certain place.  
c. *John is at home.*  $\rightarrow$   
[event *BE* ([thing JOHN], [place AT ([thing HOME])])]
- (20) a. *STAY* (thing, place)  $\rightarrow$  event  
b. A thing stays in a certain place.  
c. *John remains at home.*  $\rightarrow$   
[event *STAY* ([thing JOHN], [place AT ([thing HOME])])]

Notice the use of the function *AT* in examples (19c) and (20c) which maps to a place. The use of place functions such as this, plus property functions, will be explained in section 3.2.4. The two functions above is similar in the sense that the thing involved (the first argument to both) does not undergo any change. Instead, it is just at a given place in pattern (19) and stays (does not leave) this place in pattern (20). The only difference, really, is that *BE* maps to a state, while *STAY* maps to an event.

- (21) a. *GO* (thing, path)  $\rightarrow$  event  
b. A thing goes on a locative path.  
c. *John went away.*  $\rightarrow$  [event *GO* ([thing JOHN], [path AWAY])]
- (22) a. *EXT* (thing, path)  $\rightarrow$  state  
b. This is similar to *GO*, but instead, different parts of the thing occupy different parts of the path, all at once. A different way of saying it is that the thing extends across the path.  
c. *The road goes across the river.*  $\rightarrow$   
[event *EXT* ([thing ROAD], [path ACROSS ([thing RIVER])])]
- (23) a. *ORIENT* (thing, path)  $\rightarrow$  state

- b. A thing is oriented along a path, but it neither travels nor occupies it.
- c. *The sign points across the river.* →  
[event ORIENT ([thing SIGN], [path ACROSS ([thing RIVER]])])]

The functions in (21), (22) and (23) above all involve a thing and its relation to some path. The difference between them is that *GO* constitutes an event where the thing actively moves along the path, while *EXT* and *ORIENT* are instead states saying something about the thing's position according to the path.

- (24) a. MOVE (thing) → event
  - b. A simple function expressing the event of a thing moving.
  - c. *Debbie danced.* → [event MOVE ([thing DEBBIE])]
- (25) a. CONF (thing) → state
  - b. Similar to *MOVE*, this function instead expresses a state where the thing is in a certain position.
  - c. *Sally sat still.* → [event CONF ([thing SALLY])]

The functions *MOVE* and *CONF* [15, p. 89-91] are similar to each other in that they involve a thing's movement, or not, and without a known path. As such, they are similar to the functions *GO* and *BE*, respectively, except that the second argument, the path, is removed.

Patterns (26) and (27) below show examples of functions that map to paths, rather than events or states. A multitude of other path functions are also available, such as *TOWARD*, *AWAY-FROM* and *VIA* [15, p. 46-47].

- (26) a. FROM (thing/place) → path
  - b. This specifies a path starting from some thing or some place. It effectively expresses the thematic role Source.
  - c. *Max traveled from England.* →  
[event GO ([thing MAX], [path FROM ([place ENGLAND]])])]
- (27) a. TO (thing/place) → path
  - b. This specifies a path that ends when it's at some thing or some place. It expresses the thematic role Goal.
  - c. *Max traveled to Spain.* →  
[event GO ([thing MAX], [path TO ([place SPAIN]])])]

### 3.2.2 Aspectual Functions

Next we have the aspectual functions. These express two different aspects one can have on a situation, namely inchoative and perfective aspects. Jackendoff [15] only had one of these, namely *INCH* [15, p. 75], but later revisions to the theory added to function *PERF* [13, p. 364], which is actually sort of the opposite of *INCH*. Both are explained in patterns (28) and (29).

- (28) a. INCH (state)  $\rightarrow$  event  
 b. Expressing the inchoative, this function takes a state and returns an event, effectively giving the event of that state coming about.  
 c. *John ended up at home.*  $\rightarrow$   
 [event INCH ([state BE ([thing JOHN],  
 [place AT ([thing HOME]))])] ]
- (29) a. PERF (event)  $\rightarrow$  state  
 b. The opposite of *INCH*, this function expresses the perfective, taking an event and returning a state. It essentially gives the state of the event being completed.  
 c. *Mary has gone into the house.*  $\rightarrow$   
 [event PERF ([event GO ([thing MARY],  
 [place IN ([thing HOUSE]))])] ]

### 3.2.3 Causative Functions

The causative functions all express events involving a cause and an effect. Jackendoff [15] only used the standard *CAUSE* function, but later revisions [13, p. 364] added the functions *LET* and *HELP* as well. All three are shown below.

- (30) a. *CAUSE* (thing/event, event)  $\rightarrow$  event  
 b. A thing or an event causes another event, which can be called the effect. Note that Jackendoff [13, p. 364] has an alternative version of this function with three arguments, with the additional being another thing. This version should be used if a Patient is known in the event (the new thing argument). The same principle for this alternative version is also applied to the *LET* function below. This chapter will return to the issue of thematic roles in relation to conceptual structures in section 3.6.  
 c. *The wind made it rain.*  $\rightarrow$   
 [event *CAUSE* ([thing WIND], [event RAIN])] ]
- (31) a. *LET* (thing/event, event)  $\rightarrow$  event  
 b. This function is very similar to *CAUSE*, but it involves verbs such as *allow* and *let*, where the causer indirectly makes an effect come about by allowing or letting it happen.  
 c. *Mary allowed the party to happen.*  $\rightarrow$   
 [event *LET* ([thing MARY], [event PARTY])] ]
- (32) a. *HELP* (thing, thing, place)  $\rightarrow$  event  
 b. Also similar to *CAUSE*, this function relates to events where the first argument helps, aids or assists in making the effect happen. As such, the event may not finish. Notice that this function only has a three argument version, as opposed to *CAUSE* and *LET*.

- c. *Beth assisted Harry in washing the dishes.* →  
 [event HELP ([thing BETH], [thing HARRY],  
 [event WASH ([thing DISHES])])]

### 3.2.4 Other Functions that Map to Semantic Concepts

Many different functions may map to the other semantic concepts mentioned. Place functions typically take a thing as an argument and expresses some location according to this thing. Examples include *IN*, which expresses the place inside some thing, *AT*, which expresses the location at the thing, and *AT-END-OF*, which specifies the place at the end of the thing.

Property functions can be basically any adjective, and the functions typically take no arguments. Examples include *GREEN*, *BIG* and *HEALTHY*. Time functions relate to time adjuncts, such as *at noon* and *on Monday*. These functions also typically take no arguments, and other examples are *TUESDAY* and *YESTERDAY*.

### 3.2.5 Modifying Functions

For these kind of functions the main purpose is to convert from one state or event into another state or event, making that initial concept a conceptual restrictive modifier of the whole structure. Because of this special behaviour the notation will exclude the name of the semantic concept marker for these functions, since they are the same as the overall structure. The functions are added into a conceptual structure, adding further information, using the notation [*<MODIFYING-FUNCTION>* [event/state]]. The most important modifying function is perhaps *BY* [15, p. 96], explained in pattern (33).

- (33) a. [BY [event/state]]  
 b. This expresses the means by which an event happens, where, for instance, someone does an event *by means of* some other event.  
 c. *John went into the room by going through the window.* →  
 [event GO ([thing JOHN],  
 [path TO ([place IN ([thing ROOM])])])]  
 [BY [event GO ([thing JOHN],  
 [path VIA ([thing WINDOW])])]]]

Other restrictive modifying functions include *FROM*, *FOR*, *WITH* and *EXCH* [15, p. 98-99], which will be explained only briefly as their pattern is very similar to the *BY* function shown in (33). The function *FROM* defines the cause of some event usually expressed in syntax with words like *because* and *from*. *FOR* can show the goal or purpose an event is supposed to reach, for instance an event or state someone wishes to happen by doing some initial event. Next, *WITH* may express the state of something during an event. It is referred to by Jackendoff as "the subordinating function of accompaniment." Finally, *EXCH* expresses exchange, where the value of the main clause in a sentence corresponds to the value of a subordinating

clause. Example sentences for the modifier functions mentioned are shown in (34).

- (34) a. FROM: *John turned yellow from eating carrots.*  
 b. FOR: *Bill obtained the book (in order) to give it to Harry.*  
 c. WITH: *Bill entered the room with a smile.*  
 d. EXCH: *Bill gave Harold \$5 for mowing the lawn.*

### 3.3 Semantic Fields

Up until now, all the functions covered have been in the *locative* field, meaning that they've all centered around the spacial movement or configuration of some thing. But Jackendoff also proposes three additional fields, namely the *temporal*, *identificational* and *possessional* fields [15, p. 26, 135] [26, p. 282]. Essentially, the fields are used to represent subcategories of the semantic concepts. So, for instance, instead of just states we have locative states, possessive states and so on. The temporal field involves time, the identificational involves properties and the possessional involves ownership.

The semantic fields may constrain (or decide) the type of the arguments that the functions can take. For example, as seen, the locative *BE* function must have as its second argument a physical place or location. The fields are shown in the notation as subscripts to the semantic functions; *LOC* for locative, *IDENT* for identificational, *POSS* for possessive and *TEMP* for temporal. So, the simple *BE* and *GO* functions explained earlier should actually have had the *LOC* subscript added, resulting in *BE<sub>LOC</sub>* and *GO<sub>LOC</sub>*.

Examples (35) and (36) show structures using the three additional fields for the functions *BE* and *GO* (for examples of the locative use, see patterns (19) and (21)).

- (35) a. *The party is on Saturday.* →  
 [state BE<sub>TEMP</sub> ([thing PARTY], [place AT ([time SATURDAY])])]  
 b. *The theatre is full.* →  
 [state BE<sub>IDENT</sub> ([thing THEATRE],  
 [place AT ([property FULL])])]  
 c. *This book belongs to John.* →  
 [state BE<sub>POSS</sub> ([thing BOOK],  
 [place AT ([thing JOHN])])]
- (36) a. *The party has been moved to Saturday.* →  
 [event GO<sub>TEMP</sub> ([thing PARTY],  
 [path TO ([place AT ([time SATURDAY])])])]  
 b. *Mary went from being depressed to being happy.* →  
 [event GO<sub>IDENT</sub> ([thing MARY],  
 [path FROM ([place AT  
 ([property DEPRESSED])]),  
 TO ([place AT ([property HAPPY])])])]

- c. *The prize went to Kate.* →  
 [event GO<sub>POSS</sub> ([thing PRIZE],  
                   [path TO ([place AT ([thing KATE])])])]

Notice that the field subscript is only applied to the outer most function, which implies that the field also counts for the inner structure. As such, the inner functions (*AT*, for instance) could also have the subscript added, but it is removed since the existence of it in the surrounding function makes it all-encompassing. Notice also that the nature of the arguments differ depending on which field is used. For instance, in example (35a) the place function *AT* takes a time argument, but in example (35b) it takes a property argument.

### 3.4 Semantic Features

This chapter has already mentioned that *CAUSE* may have additional features attached to specify further meaning to the function. Such semantic features are added to functions as superscripts. If there are more than one feature, they will be separated by commas. Jackendoff proposes three possible features to the *CAUSE* function, specifying whether the effect is successful, using +, or not, using -, or if it is undetermined, using *U* [15, p. 132]. A feature is also proposed to capture whether the Agent (or causer) moves along in the effect or not. If it does the feature *entrain* (with the superscript *E*) is added, and if not the feature *launch* (using the superscript *L*) is added [15, p. 138]. The examples in (37) show how the two features can be used. Note that Jackendoff originally added *entrain* and *launch* as subscripts to the function, but this author feels it is more appropriate to add them as superscripts. This is also the case with most of the other semantic features mentioned below.

- (37) a. *Bill dragged the car down the road.* →  
           [event CAUSE<sup>+,E</sup> ([thing BILL],  
                               [event GO ([thing CAR],  
   [path DOWN ([thing ROAD])])])]
- b. *Bill threw the ball into the field.* →  
           [event CAUSE<sup>+,L</sup> ([thing BILL],  
                               [event GO ([thing BALL],  
   [path FROM ([thing BILL])  
   TO ([place IN ([thing FIELD])])])])]

In example (37a), Bill successfully makes the car go down the road, and he follows, or takes part, in the effect. Bill also successfully makes the ball go into the field in example (37b), but he does not take part in the effect (he does not end up in the field as well). Notice also the use of both *FROM* and *TO* as path functions in (37b). This is an example of a conjunctive structure of several functions that produces just one semantic concept.





- (41) a. +B and +I : Telic, iterative event, such as *The light flashed until dawn*.  
 b. +B and -I : Telic, non-iterative event, such as *John ran to the shop*.  
 c. -B and +I : Atelic, iterative event, such as *The light was flashing*.  
 d. -B and -I : Atelic, non-iterative, such as *John ran*.  
 e. *The light flashed*  $\rightarrow$  [event LIGHT-FLASHED<sup>+B,-I</sup>]

### 3.5 Combinatory Functions

The following functions are given semantic concepts, with specific semantic features, as arguments and produce the same type of concepts, but change the nature of the features. Often, they require the arguments to have specific values for their semantic features. The functions are mostly used for things, but might also be used for the other categories, especially events. There are two kinds of combinatory functions, namely *including* functions and *extracting* functions [26, p. 286-288].

#### 3.5.1 Including Functions

These functions add the whole argument to the construction, effectively "including" it in the structure. They are the functions *PL*, *COMP* and *CONT*, shown below. Note that <SC> stands for any semantic concept, but for these functions it is usually either a thing or an event. The value at the right side of the  $\rightarrow$  symbol constitutes the concept that is returned by the function.

- (42) a.  $PL (<SC>^{+B,-I}) \rightarrow <SC>^{-B,+I}$   
 b. The *plural* form of the given semantic concept. It takes a bounded argument without an internal structure and produces an unbounded argument that has an internal structure.  
 c. *Cars* has the structure [thing  $PL^{-B,+I}$  ([thing  $CAR^{+B,-I}$ ])]
- (43) a.  $COMP (<SC>^{-B,-I}) \rightarrow <SC>^{+B,-I}$   
 b. The concept produced is *composed* of the argument concept. This argument must be unbounded and have no internal structure, and the result is a concept that is bounded, but still without an internal structure.  
 c. *House of wood* has the structure [thing  $HOUSE COMP^{+B,-I}$  ([thing  $WOOD^{-B,-I}$ ])]
- (44) a.  $CONT (<SC>) \rightarrow <SC>$   
 b. The concept produced *contains* the argument concept. There are no requirements for semantic features and they are not changed.  
 c. *Chicken curry* has the structure [thing  $CURRY CONT^{-B,-I}$  ([thing  $CHICKEN^{-B,-I}$ ])]

### 3.5.2 Excluding Functions

These functions extract parts of the argument into the overall construction. They include *ELT*, *GR* and *PART*, shown below.

- (45) a.  $ELT (<SC>^{-B,+I}) \rightarrow <SC>^{+B,-I}$   
 b. The concept produced is a specific *element* of the argument concept. The function creates a bounded concept without an internal structure from an unbounded concept with an internal structure.  
 c. *A grain of salt* has the structure  
 $[thing\ ELT^{+B,-I} ([thing\ SALT^{-B,+I})]]$
- (46) a.  $GR (<SC>^{-B,-I}) \rightarrow <SC>^{+B,-I}$   
 b. This function is called the *universal grinder*. It is used for when count nouns are used as mass nouns. Therefore, the argument must be bounded, and the result is an unbounded concept.  
 c. *Dog* in the sentence *There was dog all over the place* has the structure  $[thing\ GR^{-B,-I} ([thing\ DOG^{+B,-I})]]$
- (47) a.  $PART (<SC>) \rightarrow <SC>$   
 b. This function takes an individual as argument and gives a *part* of that individual, hence the semantic features remain the same.  
 c. *The leg of the table* has the structure  
 $[thing\ LEG\ PART^{+B,-I} ([thing\ TABLE^{+B,-I})]]$

## 3.6 Thematic Roles in Conceptual Structures

When talking about the participants (or roles) of a sentence, Jackendoff uses both a thematic tier and an action tier that represent the thematic and the action dimensions, respectively [15, p. 125-151]. The list of roles are then divided under these two categories. The thematic tier includes *Agent*, *Theme*, *Goal*, *Source* and *Location*, while the action tier includes *Actor*, *Experiencer*, *Patient*, *Beneficiary* and *Instrument*.

The examples explored up until now have all had structures that represent the thematic tier, but Jackendoff also includes functions that represent the action tier. These are *AFF*, for affection, and *REACT* for reaction. These functions act as supplements to the original structures and shows the relationship between two things. As such, both take two things as arguments.

The *AFF* and *REACT* functions have one semantic feature which concerns volition. This can have a positive, negative or unknown value, which is represented as the superscripts +, - and *U* (unknown), respectively. The + superscript says that the first argument affects or reacts to the second argument intentionally, while - says the opposite.

The thematic roles are directly related to the arguments to the semantic functions. The Theme is always the first argument to the *GO*, *STAY*, *BE*,

*ORIENT, EXT, MOVE* and *CONF* functions. The Source and Goal, as we've seen, relates to the path functions *FROM* and *TO* [15, p. 46-50]. The Agent is the first argument to the *CAUSE* function, and the role Experiencer is any argument to a state function that involves mental states [15, p. 262]. The Instrument is typically an argument inside a *BY* function [15, p. 142]. Also, the first argument to *AFF* is the Actor, while the second is the Patient if the *AFF* function is not volitional, and the Beneficiary if it is [15, p. 128, 133-137]. These roles are reversed if the *REACT* function is used instead. Note that Jackendoff believes that thematic roles should not be used directly in semantic structures, but are rather "relational notions defined structurally over conceptual structure" [15, p. 47].

### 3.7 Inference Rules

The main approach Jackendoff uses when it comes to verbal inference phenomena involves the existence of structures inside other structures. Essentially, a conceptual representation entails every structure that is embedded inside it. Inferences involving causative and inchoative sentences, an example of which can be seen in (48), are handled by this principle.

- (48) a. Brutus killed Caesar.  
 b. [event CAUSE<sup>+</sup> ([thing BRUTUS],  
                                 [event INCH ([state BE<sub>IDENT</sub> ([thing CAESAR],  
   [place AT ([property DEAD])]])])] ]]  
 c. Caesar died.  
 d. [event INCH ([state BE<sub>IDENT</sub> ([thing CAESAR],  
                                 [place AT ([property DEAD])]])] ]]  
 e. Caesar is dead.  
 f. [state BE<sub>IDENT</sub> ([thing CAESAR],  
                                 [place AT ([property DEAD])])]

The structure for sentence (48c), seen in (48d), exists inside the structure for sentence (48a), shown in (48b). Because of the use of the inchoative event as an argument to the *CAUSE* function (it is the effect of the causative event), structure (48b) entails structure (48d). Similarly, structure (48d) entails structure (48f) because the state of Caesar being dead (the structure of which is shown in (48f)) is embedded inside the inchoative event, being used as an argument to the *INCH* function.

Conceptual structures employing conjuncted functions also entail the structures where one or more of these functions are removed. This applies to the structures seen in (49), for instance, where structure (49b) entails structure (49d) because the latter is the same structure as the former, but without the place-function.

- (49) a. *Jane danced at the club.*  
b. [event MOVE ([thing JANE]) [place AT ([thing CLUB])]]

- c. *Jane danced.*
- d. [event MOVE ([thing JANE])]

In addition, more specified structures entail structures that leave one or more arguments empty. For instance, the structure in (49d) would entail a structure like *[event MOVE ([thing ])]*, where the thing is unspecified. In other words, an event where Jane moves also means that there is an event where some thing moves.

So, in conclusion, semantic entailment in the theory of conceptual structures is captured by the use of embedded structures, the existence (or non-existence) of conjuncted functions, and the specificity of the semantic concepts.

### 3.8 Flat Conceptual Structures

In order to compare Jackendoff's conceptual structures to the two other kinds of representation that will be presented in the following chapters, there might be times when alternative forms for the structures will be considered. Since Jackendoff's theory focuses on embedded structures, this alternative formation instead groups all the functions used side by side, in a conjunctive manner. This results in structures that are much more similar to the formulas used by the other theories, which are typically not embedded, but conjunctive. So, when discussing various entailment examples, these so called "flat" structures will sometimes be used, in order to better compare the three different approaches.

The flat structures will basically consist of the same semantic functions used in the regular embedded structures, but with some added functions that constitute the nature of the arguments to the functions. Variables will also be used for all semantic concepts that act as arguments, such as events and things. The example in (50) serves as an illustration of how such flat structures look like. The structure in (50b) shows the normal conceptual structure for the sentence in (50a), while structure (50c) shows the flat structure version. Notice that the latter consists of conjuncted functions (separated by the symbol &) rather than being an embedded structure.

- (50) a. *Mike is at home.*
- b. Conceptual structure:  
          [state BE<sub>LOC</sub> ([thing MIKE], [place AT ([thing HOME])])]
- c. Flat conceptual structure:  
          [BE<sub>LOC</sub>(s) & ARG1(s, t) & ARG2(s, p) & MIKE(t) & AT(p) & ARG1(p, t<sub>2</sub>) & HOME(t<sub>2</sub>)]

The functions that are used include the BE<sub>LOC</sub>, AT, MIKE and HOME functions, which are also used in the normal conceptual structure. These four functions all represent the nature of the various variables that operate as their respective arguments. These variables therefore stand for all the specific semantic concepts used in the structure, which include two things (*t* and *t*<sub>2</sub>), a state *s*, and a place *p*. The semantic functions thereby say that

the  $t$  variable is an instance of a *MIKE*,  $t_2$  is an instance of a *HOME*,  $p$  in an instance of an *AT* place, and  $s$  is an instance of a locative *BE* state.

In addition, the flat structures make use of argument functions that represent which variables certain other functions take. These are labeled  $ARGNn$ , where  $n$  can be any number that represents the argument place for the original function used in the normal conceptual structure. The first argument to the *ARG* functions refers to the semantic concept that takes the given argument, and the second argument is the the argument itself.

So, for instance,  $ARG1(s, t)$  in the example says that the first argument to the  $BE_{LOC}$  function (which represents the state  $s$ ) is the variable  $t$ . Notice that a  $BE_{LOC}$  function takes two arguments, so in the flat structure the functions  $ARG1$  and  $ARG2$  are used to specify these. Similarly, the *HOME* function takes no arguments, so no argument function that uses  $t_2$  as its first argument is present.

As a final comment on these flat structures, it should be understood that all the variables used are introduced via existential quantifiers, which indicate that they are available and present in the whole structure. Still, a choice has been made not include the quantifiers, simply to make them a bit more readable. Although the semantic representation that will be introduced in chapter 5 also does not directly include such universal quantifiers, the theory in chapter 4 will, on the other hand.

Note also that because of the conjunctive nature of the flat structures, there might exist problems with the scope of certain functions, in accordance with the quantified variables. So, the flat structures are only meant as an additional illustration that can serve to make the capturing of entailments a little bit clearer.



## Chapter 4

# Parsons' Underlying Event Theory

### 4.1 Overview

"[S]imple sentences of English contain subatomic quantification over events [and states]. [...] I call it an "underlying" quantification."

— Terence Parsons, [25]: IX

Terence Parsons theory of subatomic semantics is based on the idea of underlying events and states. Essentially, behind every sentence there is a hidden event or state and the sentence in question is simply the surface realization of that kind of situation. The theory involves using formulas built up by the conjunctions of predicates, functions and atomic semantic terms, used in a framework of standard first order logic. According to Parsons, verbs are defined as "kinds of actions or states" [25, p. 4], and the existence of either of the two are examples of those kinds taking place. Variables signifying either events or states are used as arguments to semantic predicates and are thereby specifically defined. The version of Parsons' theory that this thesis will be focusing on is Parsons [25]. The theory itself is based largely on the work of Donald Davidson, for instance Davidson [5].

Because of the theory's implementation of first order logic, typical formulas expressing situations involve the use of quantifiers which introduce variables for situations. The symbol  $e$  is used for some event, and the symbol  $s$  for some state. The various participants of the situation are then identified. These may conform to the syntactic (or surface) structure of the situation, where participants are labeled as either the subject or as an object, or to the semantic structure, where instead the proper thematic roles are used as labels. The example in (51b) shows a very simple structure for the event sentence in (51a), while structure (51d) represents the state sentence shown in (51c), both utilizing the latter semantic approach. A syntactic approach would involve using the predicates *Subject* and *Object* instead of *Agent* and *Theme*.

- (51) a. *Brutus killed Caesar.*

- b.  $(\exists e)$  [Killing( $e$ ) & Agent( $e$ , Brutus) & Theme( $e$ , Caesar) & Cul( $e$ , before now)]
- c. *Caesar was happy.*
- d.  $(\exists s)$  [Being-Happy( $s$ ) & Theme( $s$ , Caesar) & Hold( $s$ , before now)]

Notice the use of the  $e$  and  $s$  symbols, which are first introduced by the use of the existential quantifier. They are then used as arguments to the predicates inside the actual formulas, which are inside square brackets. The formulas themselves are flat conjuncts consisting of a number of semantic predicates, a departure from the embedded structures used by Jackendoff in chapter 3. The *Killing* predicate says that the situation in question is an event of killing, where an Agent and a Theme is known to be participating. The *Being-Happy* predicate, on the other hand, represents the fact that the given state is a state of being happy.

The thematic role predicates all take two arguments (similar to most of the other predicates that will be looked at later), where the first argument constitutes the given event or state, and the second argument is typically a constant atomic symbol. This symbol is most often the same as the syntactic form that the participant uses in the surface structure (in this case *Brutus* and *Caesar*). The use of thematic roles will be explained in more detail in section 4.2.

The *Cul* and *Hold* predicates are used to represent tense, or when the situations happens. In this case, both occurred before the present time. This kind of representation will become a little more complex later on, as Parsons instead opts for using time variables, similar to the use of the  $e$  symbol, which are then related to time constants (such as *now*). The difference between *Cul* and *Hold* will be explained later, as well, in section 4.4. So, the *before now* value used here is only a temporary method of referring to the past.

To summarize the structures in (51), the *Killing* predicate represents the type of verb used to identify the event, the *Being-Happy* represents the type of adjective used to make the state, the *Agent* and *Theme* predicates the thematic roles (participants) and the *Cul* and *Hold* predicates the tense of the sentences. This kind of representation is more detailed compared to a strictly logical representation, in which the structures might simply be *Kill(b, c)* and *Happy(c)*, where  $b$  stands for *Brutus* and  $c$  stands for *Caesar*.

The patterns in (52) show how predicates can be used to describe the specific type a situation has. For predicates of events, the pattern in (52a) is used. This implies that the noun form of the given verb, created by adding *-ing*, is typically the preferred form of the predicate. States, on the other hand, uses the pattern shown in (53), where the adjective is the one used in the surface sentence.

- (52) a. <Verb>-ing( $e$ ) - Example: *Killing(e)*
- b. Being-<adjective>( $s$ ) - Example: *Being-Happy(s)*

Various other predicates and functions may be applied to capture sentence meaning. The following sections will go into more detail



regarding the capturing of participants (section 4.2), modifiers (section 4.3) and the representation of tense (section 4.4). Section 4.5 will focus on the issue of causative and inchoative sentences, while section 4.6 will look into the way the progressive and perfect aspects are represented. Finally, section 4.7 will explain the way the theory captures and maintains various inference phenomena.

## 4.2 Thematic Roles

When specifying the participants of situations, the pattern <Thematic-Role>(e/s, X) is used, where X can be any term that represents a particular participant. Parsons uses the thematic roles *Agent*, *Theme*, *Source*, *Goal*, *Benefactive*, *Instrument*, *Performer* and *Experiencer* for events, and *Performer*, *Experiencer* and *Theme*, and possibly also *Instrument*, for states [25, pp. 73-74].

The role of Performer might need some explanation. Parsons uses this alternative to the role of Instrument for sentences where a participant, which might appear to be the Instrument, really shouldn't be labeled as such. Consider the sentence *The knife cut Paul's leg*, for instance. Here, it is logical to assume that *the knife* is an Instrument, which is always used or controlled by some Agent. But it might not be the case that someone actually used the knife intentionally; Paul might have just stepped on the knife and cut himself. Therefore, the role of Performer is more preferable in this case, as it implies a participant which performs an action, but which has no immediate agency and is not controlled by anything.

Parsons also proposes the use of combined thematic roles, specifically the roles *Agent-Theme*, *Performer-Theme* and *Experiencer-Theme*. These three are all alternatives to the Theme role, but which also have the additional qualities of Agent, Performer or Experiencer. In addition, the role of *Agent-Performer* constitutes a participant which is both an Agent and a Performer [25, pp. 81-82]. If either of these combined roles appear in a situation, none of the roles that make up the combination can also appear. For instance, if there is an Agent-Theme present then no other participant can be an Agent or a Theme, as the combined role occupies both of these role slots. This essentially follows the view that a situation only allows one instance of each of the thematic roles.

As a final illustration, the table below shows the entire list of roles, with examples and explanations as to when they typically appear. The examples have their respective roles labeled with a subscript, which is the first letter of the given role.

Thematic Role:	Example and typical appearance:
Agent	<i>Brutus<sub>A</sub> killed Caesar.</i> Subject in causative sentences.
Theme (Patient)	<i>Brutus killed Caesar<sub>T</sub>.</i> Object in causative sentences; Subject in copular sentences.
Source	<i>He travelled from Norway<sub>S</sub>.</i> Indirect object; Argument in <i>from</i> - phrases.
Goal	<i>He travelled to France<sub>G</sub>.</i> Indirect object; Argument in <i>to</i> - phrases.
Benefactive	<i>He gave her<sub>B</sub> a gift.</i> Indirect object; Argument in <i>for</i> - phrases.
Experiencer	<i>She found the book.</i> Subject.
Instrument	<i>She wrote the letter with a pen<sub>I</sub>.</i> Argument in <i>with</i> - phrases; Subject.
Performer	<i>The wind<sub>P</sub> opened the door.</i> Subject; Non-agentive arguments in <i>by</i> -phrases.
Agent-Theme	<i>Mary<sub>AT</sub> runs.</i> Both Agent and Theme.
Performer-Theme	<i>The statue<sub>PT</sub> stood in the corner.</i> Both Performer and Theme.
Experiencer-Theme	<i>She<sub>ET</sub> hungers for his love.</i> Both Experiencer and Theme.
Agent-Performer	<i>The ball<sub>AP</sub> followed another one down the street.</i> Both Agent and Performer.

### 4.3 Modifier Predicates

A formula may have a multitude of different modifier predicates, which are used to capture the meaning formed when adjuncts are used [25, pp. 40-41]. Traditionally there are three types of adverb-like modifiers that can be used in syntax. These are single-word adverbs, such as *probably* and *gently*, prepositional phrases, such as *in the garden*, *after six o' clock* and *with a knife*, and subordinate clauses, which has a subordinating conjunction followed by a clause. The first two can be mapped into an underlying event formula by the use of modifier predicates, while the third one involves using two or more event variables which are joined together in some way.

For a single-word adverb, a one-argument taking predicate suffices, where the pattern is the one shown in (53a). The pattern seen in (53b), on the other hand, shows what the second kind of modifier looks like, which maps prepositional phrases. The only difference is the fact that the predicate takes an additional argument (here *X*), which is the same argument that is present in the given prepositional phrase.

- (53) a. <Mod>(e/s) - Example: *Slow(e)* for a situation that is done *slowly*.

- b.  $\langle \text{Mod} \rangle(e/s, X)$  - Example:  $\text{In}(e, \text{Garden})$  for the phrase *in the garden*.

Adjectives modifying nouns, such as *red* in *a red house*, can be logically understood as the conjunction of semantic predicates. Say, for instance, that  $X$  is *a red house*, then you could also say that  $X$  is *a house* &  $X$  is *red*. However, this approach does not work with phrases such as  $X$  is *a clever teacher*, where it may not be the case that  $X$  is also generally clever (or a clever human being).

Being clever is therefore understood, according to Parsons, as being *clever for an F*, where  $F$  depends on the context, which would be *teacher* in this case. The phrase  $X$  is *a clever N* then means that  $X$  is *an N* &  $X$  is *clever for an F*. Most often,  $F$  is the same as  $N$ , but not always. This can be applied to all adjectives. The sentence *He is a tall basketball player*, for example, can be understood as *He is a basketball player* & *he is tall for an F*, where  $F$  might just be *a basketball player*, or *a basketball player in grade school* or something else [25, pp. 43-44].

Though this is an important matter, Parsons has decided to leave it in the background, relying on the use of the simple patterns seen in (53) for all kinds of verbal modifications. As such, whenever a state is involved in a modifier predicate, the notion of an  $F$  should be understood as an additional argument, even though it will not be shown directly in any formula.

The pattern in (54) shows a final illustration of how modifiers are added to formulas for situations. Note that  $\alpha$  can be either an event or a state,  $\langle \text{AdvMod} \rangle$  can be any single-word adverb modifier, and  $\langle \text{PrepMod} \rangle$  can be any preposition introducing a prepositional phrase. Note that there are also additional predicates which can take more than two arguments, such as *Between*, which places the situation between two given entities. However, the one- and two-argument taking predicates are the most common.

$$(54) (\exists \alpha) [\langle \text{AdvMod} \rangle(\alpha) \ \& \ \langle \text{PrepMod} \rangle(\alpha, X) \ \& \ \dots]$$

Parsons divides all adverbials into six different semantic categories (regardless of how many arguments they take), depending on what field of meaning they occupy. The categories include *locative* (*In*, *Under* etc.), *motion* (*Through*, *Between* etc.), *direction* (*Away*, *Down* etc.), *orientation* (*Crosswise*, *Vertical* etc.), and *manner* (*Gently*, *Slowly* etc.) adverbial predicates, plus some other *miscellaneous* kinds (like *In*, as in *in the back*, for instance) [25, pp. 269-270].

Note that many different adverbials appear in several categories. For example, the predicate *Here* is either a locative, motion or direction predicate, depending on the context. Note also that Parsons uses the predicate *With* to refer to Instruments, labeling it as its own *instrumental* category. In this thesis, on the other hand, the predicate *Instrument* will be used instead to indicate the presence of an Instrument.

## 4.4 Representing Time

When capturing the important meaning of tense, Parsons uses both variables for points in time and intervals of time [25, pp. 167-170]. Both of these are introduced in the same way as the event and state variables are, using the existential quantifier. All situation types are related to some time interval, typically labeled with the symbol  $I$ , which can have happened in the past, the present or the future. To capture this distinction, a constant symbol called *now* is used to represent the present time and the functions  $<$ ,  $>$  or  $=$  are used to relate the interval of time to the present time.

The point or points in time, which are typically labeled with the symbol  $t$ , are usually said to be inside the interval of time, using the "element of" relation  $\in$ . This essentially says that the point in time is a member of the entire set of time points which make up the interval of time. The interval of time might also be constrained within certain time periods. For example, if the time adverbial *at noon* is used, the fact that the interval of time is a subset of the entire set of noons must be encoded. This is done by using the subset symbol  $\subseteq$ .

As an example of this approach to time, consider the sentence in (55a), which has the formula shown in (55b). As can be seen by the function  $I < \text{now}$ , the sentence is in the past tense. It also happened *Yesterday at noon* which is captured by the two time interval constraints [25, pp. 208-222].

- (55) a. *Yesterday at noon, Nick ran.*  
 b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ I \subseteq \text{Noons} \ \& \ I \subseteq \text{Yesterday} \ \& \ \text{Agent-Theme}(e, \text{Nick}) \ \& \ \text{Cul}(e, t)]$

Note that there is a difference between time interval modifiers (like  $I \subseteq \text{Noons}$ ) and temporal modifiers of events (like  $\text{At}(a, \text{Noon})$ ). Basically, these two uses represent two different meanings that can appear when a time adverbial is used; one modifying the interval the event happens and the other modifying the event itself.

The *Cul* predicate says that the event in question culminates at the given time point, i. e. it finishes. The *Hold* predicate, on the other hand, represents the fact that an event or state holds or is valid for a certain time [25, pp. 23,25,171]. These two predicates are used to capture various situation types, for which Parsons adopts the traditional four category approach used by Zeno Vendler [29]. These are *accomplishments*, which last for a certain amount of time and then finish (culminate), *achievements*, which culminate instantaneously, *states*, which only hold for a given time, but do not culminate, and finally *processes*, which also do not culminate but only hold. The list in (56) shows the four types, together with the predicates used, and some examples.

- (56) a. Accomplishment: *Hold* and *Cul*. Example: *Mary climbed the mountain.*  
 b. Achievement: *Cul*. Example: *Mary reached the top of the mountain.*  
 c. State: *Hold*. Example: *John is hungry.*

- d. Process: *Hold*. Example: *John runs*.

*Cul* and *Hold* also have versions which take only one argument, signifying simply that an event or state does culminate, or holds. Further, a situation might be said to be within a specific frame which sets the context in which the whole sentence should be understood. These frame adverbials should then be put outside the whole formula, resulting in a structure that looks something like this: *Frame* [( $\exists e$ )...]. An example is the phrase *During the war*, which frames a situation within the context of some war. *At noon* can also be used as a frame adverbial, so this is an example of a phrase that can act as all the different types of time adverbials mentioned [25, pp. 211-212].

As a final illustration of the way tense is captured in Parsons formulas, consider the pattern shown in (57). The symbol  $\alpha$  can be either an event or a state. The *<TemporalModifier>* field can be any predicate that represents a temporal adverbial modifying the event. Note that the text inside the {} brackets constitute the possible values that can appear in the given position, separated by commas. The *Time-Constraint* field stands for any possible constraints on the interval of time, such as  $I \subseteq \text{Noons}$  [25, p. 209].

- (57) *Frame* [( $\exists \alpha$ )( $\exists t$ )( $\exists I$ ) [ $t \in I \ \& \ I \{<, >, =\}$  now & *Time-Constraint*( $I$ ) & {*Cul*, *Hold*}( $\alpha$ ,  $t$ ) & *<TemporalModifier>*( $\alpha$ ) & ... ]]

## 4.5 Causatives and Inchoatives

Causative sentences have a general pattern where *Some event happens that causes some other event to happen* [25, pp. 105-107]. To capture this relation between events, Parsons uses a predicate called *CAUSE*, which takes two arguments. These arguments are the two events being related to each other. The *CAUSE* predicate greatly resembles Jackendoff's function for representing causatives, which, as we've seen, is also called *CAUSE*. The formula in (58b) shows how Parsons represents the sentence seen in (58a) [25, pp. 108-109].

- (58) a. *Mary flew the kite.*  
 b. ( $\exists e_1$ )( $\exists t$ )( $\exists I$ ) [ $t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e_1, t) \ \& \ \text{Agent}(e_1, \text{Mary}) \ \& \ (\exists e_2)(\exists t_2) [t_2 \in I \ \& \ \text{Cul}(e_2, t_2) \ \& \ \text{Flying}(e_2) \ \& \ \text{Theme}(e_2, \text{Kite}) \ \& \ \text{CAUSE}(e_1, e_2)]$ ]

Notice the use of two distinct event variables, both labeled using a number subscript. The second event is introduced via an existential quantifier within the scope of another quantifier, which introduces the first event. The formula can be understood as there being an event, where Mary is the Agent, which causes another event, where the kite is the Theme. Both events take place within the same time interval, but at different points in time. *CAUSE*( $e_1, e_2$ ) represents the fact that the first event causes the second event.

Inchoative sentences involve, as we have seen with Jackendoff's *INCH* function, an event that brings about the existence of some state. As such,

Parsons uses something called *BECOME#* which is, as opposed to *CAUSE*, a function (or a predicate operator) rather than a predicate. This function effectively maps predicates of states to predicates of events. Consider example (59) as an illustration, where an inchoative sentence is given a suitable formula.

- (59) a. *The door closed.*  
 b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Theme}(e, \text{Door}) \ \& \ \text{BECOME\#}(\text{Closed})(e)]$

In order to represent the fact that the formula leads to the existence of a new state, Parsons uses something called meaning postulates [25, p. 52]. These are essentially rules that state that some things are true if certain elements are present within a given formula. Parsons proposes the meaning postulate shown in (60) [25, p. 125], which says that if an event formula contains a *BECOME#(Closed)(e)* function, a Theme *X* and it culminates, then there exists a state of being closed, which holds and also has that same *X* as its Theme. In other words, it captures the fact that if a door closed then there was (or is) a state of that door being closed, and there was a state of the door not being closed right before the event happened.

- (60)  $\text{BECOME\#}(\text{Closed})(e) \ \& \ \text{Cul}(e) \ \& \ \text{Theme}(e, X) \rightarrow$   
 $(\exists s) [\text{Being-Closed}(s) \ \& \ \text{Hold}(s) \ \& \ \text{Theme}(s, X) \ \& \neg \text{PREVIOUSLY}(\text{Hold}(s))]$

The predicate  $\neg \text{PREVIOUSLY}$  says that the fact that *s* holds was not true prior to the given time, since a closing event of a door cannot happen unless the door was not closed prior to the event happening. To put it more bluntly, the  $\neg \text{PREVIOUSLY}(\text{Hold}(s))$  predicate involves the specifics shown in (61) below [25, p. 119]. Note that the point in time, *t*, is included in this formula.

- (61)  $\neg \text{PREVIOUSLY}(\text{Hold}(s, t)) \rightarrow \neg(\exists t_2) [t_2 < t \ \& \ \text{Hold}(s, t_2)]$

In other words, if a state did not hold prior to the time in question (*t*), then there does not exist a point in time (*t*<sub>2</sub>) before *t* where the state holds. Removing the  $\neg$  symbol would obviously mean the opposite; that there was a point in time before in which the state held. Assuming all inchoative instances have meaning postulates similar to this one, the use of *BECOME#* will always entail the existence of some new state, given the right circumstances.

Any predicate modifying the event that acts as the argument to the *BECOME#* function can be added. This has the effect of altering the nature of the state that is produced. For example, if the word *partway* is added to the sentence in (59a) the state that comes about is not just a state of *being-closed*, but a state of *being-partway-closed* [25, p. 122].

Situations which are both causatives and inchoatives, such as the sentence in (62a) involves the use of both the *CAUSE* predicate and the *BECOME#* function. The formula for the sentence in (62a) is shown in (62b).

It essentially consists of the structure in (59b), but it is embedded inside a causative event formula, where *Mary* is the Agent.

- (62) a. *Mary closed the door.*  
 b.  $(\exists e_1)(\exists t_1)(\exists I) [t_1 \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e_1, t_1) \ \& \ \text{Agent}(e_1, \text{Mary})$   
 $\ \& \ (\exists e_2)(\exists t_2) [t_2 \in I \ \& \ \text{BECOME\#}(\text{Closed})(e_2) \ \& \ \text{Cul}(e_2, t_2) \ \& \ \text{Theme}(e_2, \text{Door})]$

## 4.6 Progressive and Perfect Aspects

In this section, Parsons' approach to the perfect and the progressive aspects will be explained. Any situation that culminates results in a resultant state. This state is different from the state that comes from inchoative sentences, which are also called "target" states. This is because resultant states are always permanent, while target states may only be temporary. For example, the resultant state of a sentence like *Paul threw the ball on the roof* is the state of the ball having been thrown on the roof, while the target state is the state of the ball being on the roof. The perfect aspect thereby involves this sort of resultant state, meaning that such a state comes into being whenever a perfect sentence is used.

Similarly, progressive sentences involve an in-progress state, i. e. a state of an event's action taking place. If the sentence is in the present tense this in-progress state holds at the current time. Past progressives, on the other hand, would have held at some time prior to now. This principle of tense also applies if the perfect form is used [25, pp. 233-236].

When representing the resultant state (*R-state*) and the in-progress state (*IP-state*), Parsons uses two different functions that map from each event or state into the given *R-state* or *IP-state* [25, pp. 259-260]. He calls these functions simply *r* and *p*, but this thesis will use the more illustrative symbols *PERF* and *PROG* instead. Both of these may be combined to form perfect progressive sentences as well. The sentences and formulas in (63) are some examples, where formula (63b) represents the present perfect sentence in (63a), formula (63d) represents the present progressive sentence in (63c), and formula (63f) represents the present perfect progressive sentence in (63e) [25, pp. 234-236].

- (63) a. *Mary has run.*  
 b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I = \text{now} \ \& \ \text{Hold}(\text{PERF}(e), t) \ \& \ \text{Running}(e) \ \& \ \text{Agent-Theme}(e, \text{Mary})]$   
 c. *Mary is running.*  
 d.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I = \text{now} \ \& \ \text{Hold}(\text{PROG}(e), t) \ \& \ \text{Running}(e) \ \& \ \text{Agent-Theme}(e, \text{Mary})]$   
 e. *Mary has been running.*  
 f.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I = \text{now} \ \& \ \text{Hold}(\text{PERF}(\text{PROG}(e)), t) \ \& \ \text{Running}(e) \ \& \ \text{Agent-Theme}(e, \text{Mary})]$

Notice the use of embedded functions inside the *Hold* predicates. Essentially, this produces states that hold at some time (as in *Hold(s, t)*,

where *s* is the event's resultant or in-progress state). So, instead of using *Cul* to say that the events culminate, we extract the events' *R*- or *IP*-states (using the *PERF* and *PROG* functions) and use *Hold* to say that they hold at the time of *t*. The Theme of the event (Agent-Theme in this case) is also always the same as the Theme of the event's *R*- or *IP*-state. For the slightly complex structure in (63f), the state in question is "the *R*-state of the *IP*-state of *e* [which] holds now" [25, p. 236].

So, different kinds of aspects are captured by the use of the mentioned functions. For simple active or passive sentences, none or them are used, and the *Cul* or *Hold* predicates simply take the given event or state, plus the time variable, as arguments. Perfect sentences uses the *PERF* function inside the *Hold* predicate. Progressive sentences uses the *PROG* function in the same way, and perfect progressive sentences uses a combination of the two. *Cul* is never used if the sentence is perfect or progressive.

## 4.7 Inference Rules

Now that we have an understanding of the way Parsons structures his semantic formulas, this section will look at the way verbal inference is handled. The theory first of all follows the standard first order logic approach to inference, whereby various kinds of rules proclaim that some truths lead to other truths. For example, there's the inference involving conjunctions (which typical formulas in Parsons' theory consist of) where each predicate that is present inside a conjunctive formula is also true on their own. This leads to the idea that a formula entails all other formulas which are either the same, or consist of predicates which are all also present in that initial formula. As an example of how this principle solves entailments involved with modifiers, consider the examples in (64) which show two sentences, each with their respective formulas below it.

- (64) a. *Brutus stabbed Caesar with a knife.*  
 b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e,t) \ \& \ \text{Stabbing}(e) \ \& \ \text{Agent}(e, \text{Brutus}) \ \& \ \text{Theme}(e, \text{Caesar}) \ \& \ \text{Instrument}(e, \text{Knife})]$   
 c. *Brutus stabbed Caesar.*  
 d.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e,t) \ \& \ \text{Stabbing}(e) \ \& \ \text{Agent}(e, \text{Brutus}) \ \& \ \text{Theme}(e, \text{Caesar})]$

The only difference between the sentence in (64a) and the one in (64c) is the presence of the phrase *with a knife*. In both cases it is true that Brutus stabbed Caesar, but in sentence (64a) we are given the additional information that Brutus used a knife in the event. As such, sentence (64a) entails sentence (64c), but not vice versa. This is captured in the formulas by the inclusion (or exclusion) of the Instrument predicate [25, pp. 13-14]. The formula in (64d) has the exact same structure as the one in (64b), minus the Instrument predicate, so formula (64b) correctly entails formula (64d).

In addition to the standard logic approach, Parsons also makes use of several meaning postulates, which was explained in section 4.5. In



that same section, the way entailments involved with causatives and inchoatives are handled was also mentioned. Essentially, the formulas shown in (59b) and (62b) solve the entailments shown in (65), which is a typical example of this inference phenomenon. The  $\rightarrow$  symbol indicates the entailment relation, so each sentence leads to the one that is right beneath it.

- (65) a. *Mary closed the door*  $\rightarrow$   
       b. *The door closed*  $\rightarrow$   
       c. *The door is closed.*

Formula (62b), which represents the sentence in (65a), entails formula (59b), which represents the sentence in (65b), because it includes an embedded event structure (using  $e_2$ ) which is exactly the same as the formula in (59b). The entailment between sentences (65b) and (65c) is captured by the use of the meaning postulate seen in (60), which creates a state of the door being closed.

Parsons also argues that his theory captures various other kinds of entailment phenomena, examples of which are shown in (66) [25, pp. 15-19]. The example in (66a) shows an entailment phenomenon involved with perception statements, where it is certain that Mary saw the stabbing in the first sentence, but not necessarily in the second. The sentences in (66b) show an example of the explicit versus implicit mentioning of events, where the second sentence can also be implicitly thought of as a flight, where this is explicitly mentioned in the first sentence. Finally, sentences (66c) show the use of two different subjects; the first with an event, where the Agent is *Mary*, and the other where the subject is a person, which is the same Agent as in the event.

- (66) a. *Mary saw Brutus stab Caesar* vs. *Mary saw the stabbing of Caesar by Brutus.*  
       b. *A flight over the Pole by a Norwegian* vs. *A Norwegian flew over the Pole.*  
       c. *Mary's singing broke the vase* vs. *Mary broke the vase.*

So, in example (66a) the first sentence should entail the second, but not vice versa, the sentences in (66b) should entail one another because they are both events of flying, involving a Norwegian and the Pole as the location, and in example (66c) the fact that Mary's singing broke the vase should entail the fact that Mary herself broke it.

This section will not go into detail as to how these phenomena are handled in Parsons' framework, as it is basically done using the same tools mentioned up until now. In conclusion, Parsons uses standard logical rules of inference, plus specific meaning postulates, to encode and handle verbal inference.



## Chapter 5

# VerbNet: A Lexical Database

### 5.1 Overview

"VerbNet is the largest on-line verb lexicon currently available for English. It is a hierarchical domain-independent, broad-coverage verb lexicon with mappings to other lexical resources..."

— VerbNet Webpage, [23]

This chapter will introduce VerbNet, an important lexical database which aims to capture and store knowledge of lexical semantics. Its main goal is to successfully represent the many meanings, or senses, that verbs can have, and how they behave. VerbNet also tries to organize and relate them all in a formal manner [23] [24] [28].

VerbNet categorizes verbs under an assortment of specialized verb classes. These classes originated with the work of Beth Levin [17], who devised them using different methods for differentiating verbs, for instance the nature and number of the arguments they take. In addition, VerbNet adds several new classes, as well as subclasses, to Levin's initial ones, resulting in a rich hierarchy of classes of varying specificity.

As of the newest edition, VerbNet has 274 first-level classes, covering over 5200 different verb senses. It is valuable to understand the way in which Levin brought about the initial classes, a topic that will be investigated first (in section 5.2), before introducing the actual structure and use of VerbNet in section 5.3. Finally, section 5.4 will explore three other notable lexical databases which, similar to VerbNet, aim to represent lexical semantics.

### 5.2 Verb Alternations and Verb Classes

Levin's verb classes are defined based on an array of different verb alternations (also called diathesis alternations), which are essentially different ways that verbs can behave in syntax [17, p. 1-5]. Levin emphasizes how effectively natural speakers of a language can understand and recognize the behaviour of verb arguments in the given language. They know, for instance, that the verb *break* can be both transitive and intransitive, having a

causative meaning in the former, where the cause of the intransitive meaning is explained, and an inchoative meaning in the latter, where the cause is unknown. A verb like *appear*, on the other hand, can not be used in the causative structure, though the inchoative is still possible. The sentences in the table below illustrate this. Note that the \* symbol means that the given sentence is ungrammatical.

Intransitive (inchoative):	Transitive (causative)
1. <i>The window broke.</i>	<i>The boy broke the window.</i>
2. <i>The rabbit appeared.</i>	* <i>The magician appeared the rabbit.</i>

Levin divides the various verb alternations into a few main categories. Some of these categories include alternations involved with the transitivity of verbs [17, p. 25], the arguments within the verb phrase [17, p. 45], "oblique" subject alternations, where a given subject may be replaced by another noun phrase but still be part of the whole meaning (it is hidden under the surface) [17, p. 79], alternations involved with reflexives (such as *himself*) [17, p. 84], and alternations involved with passive versus active constructions [17, p. 85].

Analyzing the ways verbs are different, and similar, in a range of possible alternations yields different patterns. Members that share patterns have been shown to share similar meanings as well, which categorize them in the same classes. To illustrate the idea, the four words *break*, *cut*, *hit* and *touch* are useful. All of them can be transitive; they can all take two arguments, a subject and an object, but they don't have much else in common. They differ in their participation in a variety of diathesis alternations [17, p. 5-11].

Levin considers three such alternations for this example, namely *middle* alternation, *conative* alternation and *body-part possessor ascension* alternation. Middle alternation involves making the direct object the subject, removing the original subject, and applying the verb. In addition, an adverb, such as *easily*, is added at the end of the sentence. Conative alternation happens when *at* is added just before the direct object. Finally, body-part possessor ascension is when possessor objects, like *Bill's arm*, have *on the* added in, as in *touch Bill on the arm*. The following tables illustrate which alternations are possible for the four verbs. Note that the → symbol does not mean entailment here, but rather the transformation from one alternation to another.

Normal use:	Middle alternation:
1. Mary cut the bread.	1. The bread cuts easily.
2. Janet broke the vase.	2. Crystal vases break easily.
3. Terry touched the cat.	3. *Cats touch easily.
4. Carla hit the door.	4. *Door frames hit easily.

Conative alternation:

1. Margaret cut at the bread.
2. \*Janet broke at the vase.
3. \*Terry touched at the cat.
4. Carla hit at the door.

Body-part possessor ascension alternation:

1. Margaret cut Bill's arm. → Margaret cut Bill on the arm.
2. Janet broke Bill's finger. → \*Janet broke Bill on the finger.
3. Terry touched Bill's shoulder. → Terry touched Bill on the shoulder.
4. Carla hit Bill's back. → Carla hit Bill on the back.

In other words, only *cut* and *break* can be used in the middle alternation, only *cut* and *hit* can be used in the conative alternation, and all except *break* can be used in the body-part possessor ascension alternation. This results in the following table, which reveal four different patterns.

Alternation:	<i>touch</i>	<i>hit</i>	<i>cut</i>	<i>break</i>
Middle:	No	No	Yes	Yes
Conative:	No	Yes	Yes	No
Body-part poss.:	Yes	Yes	Yes	No

These four patterns, one for each of the verbs, give rise to four different classes, and verbs with similar meanings often share the same pattern. For example, the verbs *crack*, *rip* and *shatter* are all related to *break*, and, interestingly, they all share the same pattern. The same goes for a number of verbs semantically related to the three other verbs in the study, as well. So, verbs fall into the same classes based on shared components of meaning.

The classes in VerbNet are based on this methodology. For instance, the *cut* pattern has resulted in a class for verbs involving cutting [17, p. 156], while the *break* pattern has resulted in a class involving breaking [17, p. 241].

### 5.3 The Structure of VerbNet

Each class in VerbNet has three distinct properties, or sections [28, p. 4,29] [24], the first one being the actual verbs that are *members* of the class. This list of verbs can be anything from very large to initially empty, the members instead being introduced in subclasses. A verb's membership in a given class indicates a particular sense of that verb, so if a verb is a member in many different classes, that constitutes several different senses of that particular verb.

The second property are the *thematic roles* associated with the class, usually consisting of roles covering the subject and an object [28, p. 30-34]. VerbNet utilizes about 30 different thematic roles, some notable

examples being *Agent*, *Theme*, *Patient*, *Topic* and *Stimulus*. Typically, each verb argument is given a specific thematic role, though there are cases where one role label may be applied to multiple participants. An example occurs when the role of *Agent* is used in classes involving communication between several entities. In a sentence like *Susan chitchatted with Rachel*, VerbNet maintains that both Susan and Rachel are *Agents*, where the latter is referred to as the *Co-Agent*.

The roles may have attached to them a large number of hierarchically ordered selectional restrictions, which limit the types that can be used within the roles [28, p. 35-36]. For example, an *animate* restriction (marked as [+animate]) confines the role strictly for animate objects (humans, animals etc.).

Finally, and most importantly, each class has a set of possible surface realizations called *frames*, which show how the verbs can behave [28, p. 35-37]. Each frame is marked with a syntactic category string, like *NP V NP* (the verb is preceded by a noun phrase and is followed by one as well), which acts as the frame's name. Further, a frame has an example sentence, plus syntactic and semantic structures, the former showing the behaviour of the thematic roles, possibly marked with restrictions, and the latter describing the meaning of the sentence. The semantic structures use a notation similar to Parsons' approach (see chapter 4), where variables for events, plus terms denoting participants, are used as arguments to conjuncted predicates.

Section 5.3.3 will go into more detail regarding VerbNet's semantic structures, as these are the structures that will be used when analyzing verbal entailment examples. However, section 5.3.1 will first go through a verb class example, which will serve as an illustration of what the individual classes look like in general, while section 5.3.2 will explain the way all the verb classes are organized in a hierarchical manner.

### 5.3.1 A Verb Class Example

The verb class *discover-84* (84 being the class number) has 13 members, including, for instance, *discover*, *figure out* and *guess*. There are three thematic roles, namely *Agent*, *Theme* and *Source* connected to this class. While the *Theme* and *Source* have no selectional restrictions, the *Agent* has two of them, namely +animate and +organization. These are organized inside square brackets, like this [+ANIMATE | +ORGANIZATION], where the | symbol (which separates the restrictions) means *or*. This means that if an argument has any of the restrictive features it is allowed to act as the given role. So for an argument to be allowed to act as the *Agent* in this case it must either be an animate object or an organization (or both). The selectional restrictions are organized in a hierarchy, where some main categories (there are 36 unique values present) include *concrete*, *time* and *location* [28, p. 36].

Furthermore, the class has six different frames, an example of which is *NP V that S*. This frame involves a subordinate clause, which is introduced by *that*. The structure in (67) shows the components in this frame, including

the example sentence used and the syntactic and semantic structures.

- (67) a. EXAMPLE: *I discovered that it made sense.*  
b. SYNTAX: Agent V Theme <+THAT\_COMP>  
c. SEMANTICS: DISCOVER(DURING(E), Agent, Theme, ?Source)

The syntax shows the positions of the utilized roles in relation to the verb, which conforms to the phrase pattern which makes up the name of the frame (*NP V that S*). For the example sentence the Agent is *I*, which is allowed (given the restrictions mentioned above) because this participant is animate. The verb is *discover*, while the Theme is the whole subordinate clause (including *that*). Notice the restriction applied on the Theme, organized inside the < > brackets. The +*THAT\_COMP* restriction says that the given Theme must be a *that*-compliment clause. Such syntax restrictions, which there are about 40 of in VerbNet, are also organized in a hierarchy, similar to the selectional restrictions used on the thematic roles.

The semantic structure shows the meaning of the frame using predicates that take a given number of arguments, separated by the conjunctive & symbol. In this case there is only one predicate, namely *DISCOVER*, which says that the event in question is an event of discovering. The arguments inside the predicate constitute that during the event (specified by the variable *E*) the Agent discovers the Theme. The Source from where the discovery was made is unknown here, which is marked using the ? symbol right before the role name. Note that *DURING* is a specific kind of time function, rather than a predicate, which returns various stages of the given event, in this case the "during" stage. Essentially, such time functions denote when the predicate in question is true. Finally, notice that predicates and functions are in all upper-case letters, while the arguments, particularly the thematic roles, are in all lower-case letters, except for the first letter. As mentioned, section 5.3.3 will explain the semantic structures in more detail.

### 5.3.2 The Class Hierarchy

The classes in VerbNet are organized in a hierarchic structure, where each class may have additional subclasses [28, p. 28-29]. These act as further specialized versions of their superclasses, introducing new attributes, such as member verbs, roles and frames, or modifying existing attributes. The initial superclasses are specified with a specific class name and number, while the subclasses have additional numbers attached which constitutes the subclass number.

For example, the verb class *run-51.3.2* has two subclasses, which are labeled *run-51.3.2-1* and *run-51.3.2-2*, respectively. The subclass *run-51.3.2-2* also has a subclass, which has the name *run-51.3.2-2-1*. Further, the subclass *run-51.3.2-2* has an additional Result role, as well as a number of other member verbs and frames. So, while the superclass has member verbs such as *bolt* and *climb*, the subclass has other members, such as *glide* and *race*. Subclasses can make use of (they inherit) all frames and roles that

are also present in superclasses. To illustrate this, the verb *jump* is used in all examples in the class *run-51.3.2*, even though the verb is only introduced in one of its subclasses. Also, the verb *run* is a member of the subclass *run-51.3.2-2-1*, so it can make use of all the frames in the classes *run-51.3.2-2* and *run-51.3.2*.

Verb members are not inherited by subclasses, on the other hand, as that would defeat the purpose of having subclasses in the first place. As an example, the verb *memorize* is a member in the class *Learn-14-2*, but this cannot be used in the one additional frame found in the subclass *Learn-14-2-1*. This frame has the syntactic structure *Agent V {of/about} Topic*, which results in sentences like *The president learned of/about a coup*. But you can't say *\*The president memorized of/about a coup*. Inheritance among verb classes thereby involve subclasses having all the frames and thematic roles that are also in their superclasses, but the member verbs are specific for each class.

### 5.3.3 VerbNet's Semantic Structures

As seen in the example in section 5.3.1, the semantic structures in VerbNet consist of various semantic predicates which are used to show the relations between events and the participants in those events [28, p. 37-41]. These predicates are joined together using the conjunct symbol & and may also be negated, by embedding the predicate inside another predicate called *NOT*.

The arguments appearing inside the semantic predicates may include the thematic roles specific to each class, an assortment of various universal constant arguments, such as *Forceful* and *Directedmotion*, and finally a few arguments that are verb specific. The latter argument type depends on the given verb, so an argument like *Form* will, for instance, be *Broken* for the verb *break* and *Bended* for the verb *bend*. In other words, the Theme ends up having a broken form in a breaking event, and a bended form in a bending event. Predicates also typically take an argument which denotes the *event* in question, or one of its stages (see below). Events are, as we have seen, marked with the *E* symbol.

Every semantic predicate usually takes as one of its arguments a time function, which specifies when the predicate is true. These time functions are based on an approach to time established by Moens and Steedman [20], which is itself based on Vendler's method of categorizing events [29]. By their accounts, events are divided into four categories, namely *culminations*, *points*, *processes* and *culminated processes*. These are defined depending on both their length and their outcome. In other words, the events are either atomic or extended, i. e. they are either instantaneous or last for some time, and either consequently result in a new state, or not. Following this principle, culminations are atomic events that lead to a new state, points are also atomic, but do not lead to a new state, and both processes and culminated processes are extended events, but whereas the former does not lead to a new state, the latter does. Separate from events are also the *state* category, which VerbNet does not refer to directly. In other words, there is no use of variables which directly denote states (like an *S* symbol), but they are indirectly represented using various semantic predicates which denote



states.

Events thereby have a general structure, called a *nucleus*. This consists of different stages, namely the starting point, the preparatory process, which may last for a given time, the culmination, which is an instantaneous point in time, and the consequent state. The time functions used by VerbNet returns the selected stages from a given event. They include the functions *START*, *DURING*, *END* and *RESULT*, which all take a single argument (the event variable). The illustration in (5.1) below shows the structure of an event in VerbNet, where the rectangle represents the whole event. Note that Moens and Steedman did not refer to a starting point in their original illustration, but it's included here since it seems logical in the VerbNet framework, where the *START* function may refer to the start of an event.

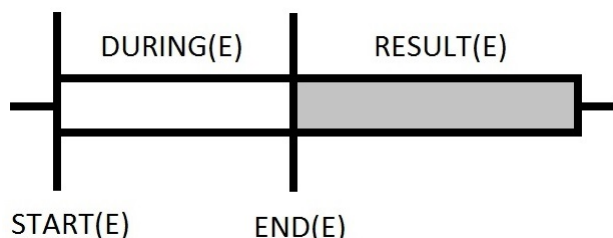


Figure 5.1: The event time structure used in VerbNet

In addition, one can also say that an event is true at all times in the event, by not using any time function. Instead, the event variable is used directly as the time argument in the given semantic predicate. For an example of the use of these time functions, the structure for the sentence *Brutus killed Caesar with a knife* can be seen in formula (68), taken from the class *murder-42.1-1*. Note the use of subscripts on some of the predicate arguments, which indicate what specific role they occupy.

- (68)  $\text{CAUSE}(\text{Brutus}_{\text{AGENT}}, E) \ \& \ \text{ALIVE}(\text{START}(E), \text{Caesar}_{\text{PATIENT}}) \ \& \ \text{NOT}(\text{ALIVE}(\text{RESULT}(E), \text{Caesar}_{\text{PATIENT}})) \ \& \ \text{USE}(\text{DURING}(E), \text{Brutus}_{\text{AGENT}}, \text{Knife}_{\text{INSTRUMENT}})$

This structure uses the predicates *CAUSE*, *ALIVE* and *USE*, plus the *NOT* predicate to indicate a negation. *CAUSE* represents the fact that the sentence is causative, where an Agent is said to cause the given event. This predicate is very similar to the *CAUSE* function used by Jackendoff (chapter 3) and the *CAUSE* predicate used by Parsons (chapter 4). Note the use of just the *E* variable in the *CAUSE* predicate here. This indicates that the fact that Brutus is the causer is true at all times in the event. *ALIVE* indicates that some participant is alive at a given stage in the event, while *USE* says that some participant uses some Instrument at some stage in the event. The *START* time function used in the *ALIVE* predicate means that Caesar was alive at the start of the event, the *DURING* function inside the *USE* predicate means that Brutus used a knife in the preparatory stage of the event, while the *RESULT* function used in the *ALIVE* predicate inside the

*NOT* predicate indicates that Caesar was not alive in the consequent state of the event.

Some event types do not have access to all the stages. Processes do not have clear start or culmination points, nor a consequent result state, so they can only refer to the preparatory stage of the event, using the *DURING* function. Culminated processes, on the other hand, do have an end point, so both *DURING* and *END* functions can be used. Points can refer to the end points and the consequent result state (*END* and *RESULT*), while culminations can refer to all stages. It is often unclear when an event starts, so the *START* function is most often used when referring to the state of something at the exact moment an event begins to happen.

*ALIVE* is an example of a semantic predicate denoting a state, saying that at some stage in an event, the state of some thing is that of being alive. Another illustrative state example is shown in (69), where a predicate curiously called *STATE* is used to denote the end state of something (the Patient in this case). Note that the role *EndState* is a verb specific argument here, similar to the *Form* argument mentioned earlier.

- (69) a. *Bill dried the clothes.*  
 b.  $\text{CAUSE}(\text{Bill}_{\text{AGENT}}, E) \ \& \ \text{STATE}(\text{RESULT}(E), \text{Dried}_{\text{ENDSTATE}}, \text{Clothes}_{\text{PATIENT}})$

In other words, the result of the event is that the state of the clothes ends up being dried. The causer of the event is Bill.

There are close to 150 different semantic predicates in the VerbNet framework and they are all divided into four main categories [23]. These are the *general* predicates, the *variable* predicates, the *specific* predicates, and the predicates for *multiple events*. The general predicates are universal across all classes, and all languages, and examples include *CAUSE* and *MOTION*. The variable predicates, which include *PREP*, *ADV* and *PRED*, are all "in a one-to-one relation with a set of words in the language" [28, p. 39]. This means that they depend on the words used in the sentence in question. Specific predicates constitute specific verbal meanings, and these predicates are each shared across a few number of verbs. For example, the predicate *SUFFOCATE* involves suffocation, and it is used by a number a verbs, where *suffocate* is only one of them.

Finally, the predicates for multiple events relate two variables to each other. They essentially represent when the events happen according to one another, so a predicate like *BEFORE*(*E0*, *E1*) would indicate that event *E0* happened before the event *E1*. Other examples include *AFTER*, which is the opposite of *BEFORE*, *CONTAINS*, which says that one event takes place inside the time interval of the other event, and *EQUALS*, which means that the two events start and end at the same points in time [28, p. 39-40].

### 5.3.4 VerbNet's Approach to Inference

Because the semantic structures in VerbNet consist of predicates and functions put together in a conjunctive fashion, the capturing of verbal

inference is similar to the approach used in Parsons' theory. Basically, a structure entails all other structures that are either the same structure or use the same predicates (with the same arguments). As an example, consider the verb *kill* again, used in the now familiar sentences *Brutus killed Caesar* and *Brutus killed Caesar with a knife*. The semantic structure for the latter sentence is shown in (68) above, while the structure for the former is the exact same, but with the *USE* predicate removed. Because of this, the latter sentence entails the former.

Another way of capturing entailment is the use the unknown symbol ? for certain roles. Consider the verb *learn*, for instance, which is a member of (among a few others) the class *Learn-14.1*. The roles available in this class are the Agent, Topic and Source. The sentence in (70a) is an example where all of these roles are present, while sentence (70c) shows a sentence where the Topic is unknown. The structure in (70b) represents sentence (70a), while structure (70d) represents sentence (70c).

(70) Verbs:

- a. *Rhoda learned French from an old book.*
- b. TRANSFER\_INFO(DURING(E), Old-Book<sub>SOURCE</sub>, Rhoda<sub>AGENT</sub>, French<sub>TOPIC</sub>) & CAUSE(Rhoda<sub>AGENT</sub>, E)
- c. *Rhoda learned from an old book.*
- d. TRANSFER\_INFO(DURING(E), Old-Book<sub>SOURCE</sub>, Rhoda<sub>AGENT</sub>, ?TOPIC) & CAUSE(Rhoda<sub>AGENT</sub>, E)

The structures in (70) make use of a predicate called *TRANSFER\_INFO*, which constitutes the knowledge of some Topic moving from a Source to some Agent. *Rhoda* is the Agent causing the event, *an old book* is the Source and the Topic is known to be *French* in structure (70b), but is unknown in structure (70d). But if Rhoda learned from an old book, then she did learn something (the Topic). Essentially, any predicate that has all of its arguments known will entail the same predicate where one or more of these arguments are missing. In other words, if Rhoda learned French, then she did learn something. Because of this, the structure in (70b) entails the structure in (70c).

In this *learn* example, the inference happens within one single verb class, but how does inference work in relation to a verb's membership in different classes? The verb *learn* is a member in both the classes *Learn-14-1* and its sister class' (*Learn-14-2*) subclass *Learn-14-2-1*. These two sister classes give the verb access to two different frames, which both make use of the *TRANSFER\_INFO* predicate. The only difference is that in the *Learn-14-1* frame, both the Topic and the Source are unknown, while in the *Learn-14-2-1* frame, only the Source is unknown. So will the entailment between these two structures be captured? If class membership is ignored, and we only consider the semantic structures, then yes, the entailment is captured (in the same way as the above example).

## 5.4 Other Lexical Resource Databases

This section will introduce three other notable lexical databases, namely WordNet, PropBank and FrameNet, which are similar to VerbNet in being lexical databases. All three will have their structure and purpose briefly explained, and how they are directly related to VerbNet in a linking system called the *Unified Verb Index* [22]. This index is an attempt to merge all of these four lexical databases together, where references between them have been added in various places. This is then a new framework, using the original, and separate, databases as starting points, but adding links between them. Examples of such links will be presented in the coming sections.

### 5.4.1 WordNet

The WordNet project was conceived by George A. Miller both to be a kind of dictionary and thesaurus, and to support data-driven applications, for use in the field of artificial intelligence, among other things [19] [18]. The database handles the four main open word classes, namely *nouns*, *verbs*, *adjectives* and *adverbs*, storing the two former in their own database and the two latter in a single one. It covers over 155 000 different word strings, or lemmas, and lists the possible senses for each of them, organized under the four classes. This representation is different from the traditional lexicon approach, where senses are separated into their own lexeme only if they are homonyms, and polysemous senses are stored under the same lexeme. In WordNet, however, all noun senses are listed under nouns, all verb senses under verbs etc., even though the senses under each class may be homonyms. To illustrate, one noun sense and two verb senses for the lemma *read* are shown in examples (71) and (72). Note that each sense has a particular number attached as a subscript, in order to distinguish it.

(71) Nouns:

- a. *read*<sub>1</sub> - Something that is read, as in *that was a good read*.

(72) Verbs:

- a. *read*<sub>1</sub> - Interpret something that is written or printed, as in *read the book*.
- b. *read*<sub>2</sub> - Have or contain a certain wording or form, as in *The passage reads as follows*.

The most important aspect of WordNet are the sets of synonyms, or *synsets*, which are collections of lemmas, each having a given sense, that represent specific lexicalized concepts. For example, the lemmas *good*<sub>12</sub>, *right*<sub>13</sub> and *ripe*<sub>3</sub> (the numbers illustrating the specific sense for each lemma) all represent the concept of *being suitable or right for a particular purpose*. Together they form a synset. In the most recent release, the database has over 117 000 such unique synsets. Each sense under a lemma points to a synset, forming a pair. *Read*<sub>1</sub> and *read*<sub>2</sub>, for instance, point to two different

synsets, namely [*read*<sub>1</sub>] and [*read*<sub>2</sub>, *say*<sub>4</sub>], forming the two pointers, or pairs, (*read*<sub>1</sub>, [*read*<sub>1</sub>]) and (*read*<sub>2</sub>, [*read*<sub>2</sub>, *say*<sub>4</sub>]), respectively.

The verb synsets are organized in a hierarchy where the links between them constitute troponymy relations (links between noun relations are hyponymy relations). All verbs are organized into two different categories, namely those that denote *actions* or *events*, and those that denote *states*. These two are further divided into more specific types, such as *manner*, *perception* and *contact* verbs for events, and abbreviations of *be* and control verbs, such as *want* and *succeed*, for states.

WordNet considers a selection of other semantic relations between synsets as well, where the most important relation is synonymy. These relations apply to all the member senses. Not all synsets may have all the relations available, though. The table below shows which relations are covered and which word class(es) they apply to, plus a few examples. The hyponymy and troponymy relations are included, as well, which make up the hierarchical organization of the synsets [19, p. 40].

Relation:	Word Class:	Examples:
Synonymy	All	<i>pipe/tube, sad/unhappy, rise/ascend</i>
Hyponymy	Nouns	<i>maple/tree, tree/plant</i>
Meronymy	Nouns	<i>brim/hat, ship/fleet</i>
Troponymy	Verbs	<i>march/walk, whisper/speak</i>
Entailment	Verbs	<i>divorce/marry, snore/sleep</i>
Causation	Verbs	<i>kill/die, show/see</i>

Note that the *troponymy* and the *causation* relations are also kinds of entailments. For instance, if someone marches then they also walk, and if someone is killed, then he also dies. The entailment relation in this table therefore denotes those entailment relations that are not causative or instances of troponymy. Note also that there is a difference between the two entailment examples given here, in that *snore* entails *sleep* directly (while someone is snoring they are also sleeping), while *divorce* entails that a *marrying* event happened at some point in time, not at the same time as the divorce event.

In addition, WordNet considers some relations that do not operate on the synsets, but rather between the words, or lemmas, themselves. This includes *antonymy*, and the *derivationally related forms*. The latter consists of the words that can be created using the morphological word root gained from the lemma in question. For instance, the verb *read* has the derivationally related forms *reader*, *reading* and *read*, which are all nouns.

Verb senses also have a list of the various ways they can be used in sentences, called the *sentence frames*. *Read*<sub>1</sub> has the sentence frames shown below, for example.

- (73) *Somebody* \_s. - Example: *He reads.*
- (74) *Somebody* \_s *something*. - Example: *She reads a book.*
- (75) *Somebody* \_s *that* *CLAUSE*. - Example: *John read that there was a robbery.*

In the Unified Verb Index, almost all member verbs in VerbNet classes refer directly to the relevant synset(s) in WordNet. For instance, the verb *advance*, which is a member of the class *Escape-51.1*, refers to the first sense of this lemma in WordNet, where it is understood as *to move forward, also in the metaphorical sense*. The fact that a member can refer to multiple synsets illustrates that there is no one-to-one relation between senses in VerbNet and WordNet; the two databases often have different ways of defining the senses.

### 5.4.2 PropBank

In contrast to both VerbNet and WordNet, PropBank, or proposition bank, covers a corpus, which is a large collection of texts where each sentence is annotated with semantic information [21] [16, p. 658-659]. PropBank uses the Penn Treebank, which annotates its sentences with their associated syntactic tree structures.

When selecting a verb in PropBank, you will find a list of the different senses available, organized under its possible variations. The verb *Read* has three different variations, namely *read*, *read up* and *read off*. Similarly, it has three different senses, one under each variation, though other verbs may have more than this. Each sense has a list of roles, or arguments, that it can assume, as well as some example sentences, all annotated with their respective role tags. The arguments are numbered as *Arg0*, *Arg1* etc., each having a brief description. Although the argument numbers usually indicate different roles depending on the given sense, *Arg0* typically denotes a volitional agent. Additional arguments which may be present, such as adverbials, are also annotated in the examples, but are not part of the core roles described. These have the notation *Argm*, plus the type of adverbial. For instance *Argm-tmp* is used for a phrase like *at night*, a time adverbial. The verb used, termed *Rel*, is also listed. Example (76) shows a prototypical entry in PropBank, illustrating the first sense of *read*, using the sentence *I read Balzac*.

(76) Arg0: I  
 Rel: read  
 Arg1: Balzac

Only two roles are realized in this sentence, although the sense has a total of four listed. The roles that are not mentioned include the *benefactive* or *direction* (*Arg2*) and the *text* or *content* (*Arg3*). Note that this example comes from the annotated corpus mentioned above.

Many, though not all, members in verb classes in VerbNet (in the Unified Verb Index) refer to the senses given to them in PropBank. For example, the verb *construct* appears in two different classes in VerbNet, where both instances refer to the same single sense in PropBank, defined as *to build or make by combining materials or parts*. The members may be related to just a single sense, or to multiple senses, similar to the mapping to the WordNet synsets. In addition, a verb appearing in many different verb

classes may, for all the membership instances, refer to the same sense(s) in PropBank.

### 5.4.3 FrameNet

The final database that will be discussed is called FrameNet, which is based on the work by the linguist Charles Fillmore and his frame semantics [1] [16, p. 659-663]. All the four major open word classes included in WordNet are also included here. In its most recent form it covers over 13000 different word senses, and also has about 190 000 different annotated sentences, which act as valuable training data in the task of semantic role labeling.

The most important elements are the *frames* (not to be confused with the frames in the VerbNet classes), of which FrameNet has about 1300. These act as structures of events, relations or entities, and also include detailed descriptions of the different participants within them. A word sense is said to evoke one or more frames, or in other words, a sense that is appropriate under a specific concept will evoke a specific frame (or frames) that covers that concept. For example, the frame *Reading* is evoked by words such as *read*, *scan* and *skim*. Example (77) shows what a typical frame looks like in FrameNet, in this case the *Reading* frame.

- (77) a. **Definition:** The *Reader* attends to a *Text* to process its information. Sometimes a particular kind of *Phenomenon* is sought in the *Text*. Example: *I READ A Tale of Two Cities last night*.
- b. **Core roles:** *Reader*, which is the one who examines a *Text* to understand it, and *Text*, which is the entity that contains linguistic symbols. The *Reader* is of the semantic type *Sentient*.  
**Non-core roles:** *Context*, which is the context wherein the *Reader* reads a particular *Text*, and *Place*, which is where the reading event takes place. *Place* is of the semantic type *Locative\_relation*.
- c. **Frame relations:** Inherits from the frame *Scrutiny* and is inherited by the frame *Reading\_aloud*.
- d. **Lexical units:** *devour.v*, *pore.v*, *read.v*, *reader.n*, *scan.v*, *skim.v*

Each frame has four main components (*a.*, *b.*, *c.* and *d.* in the example). The first is the *definition* of the frame, which explains the usage and relations of its participants in a brief and general way. Second, and perhaps most important, is the set of *elements* found in the frame. These are the participants, or roles, that can be used in the frame and they are organized under either core or non-core elements. The core elements are typically the roles that are most often present in a sentence evoking the frame, while the non-core elements are roles that are rarer. Note that not all non-core roles are shown in example (77).

In the *Reading* frame the core elements are *Reader*, the person who does the reading, and *Text*, the text that is read, and the non-core elements include roles covering such notions as the *context* in which the reading takes place, and the *time* and *place* of the reading. Each element can also have a

semantic type attached to it, which acts as a form of selectional restriction (similar to the approach in VerbNet). *Reader* must, for instance, have a *Sentient* semantic type, since only sentient beings can perform the act of reading. In addition, frame elements may include an example sentence, illustrating its use.

The third component lists important *frame relations*, expressing connections of various kinds to other frames. *Reading* has an inheritance relation to the frame *Scrutiny*, for instance, meaning that it inherits its properties from that frame. The fourth and final component is the *set of word senses*, also called the lexical units, that evoke the frame. This can include words in any of the four word classes. Note that a *v* following a word indicates that the lexical unit is a verb, while an *n* indicates a noun.

In the Unified Verb Index, the members in VerbNet classes may often refer to specific frames in FrameNet. The verb *eat*, for instance, relates to a frame called *Ingestion* in FrameNet. Similar to the mappings to synsets in WordNet and senses in PropBank, a member verb may refer to multiple frames, and instances of a member in different classes may relate to the same frame.



## Chapter 6

# Planning the Project

As these three, rather different ways of representing verbal meaning have now been introduced, the aim of the coming chapters will be to examine and discuss a variety of entailment phenomena. In doing so, the focus will be on each theory's strengths and weaknesses when it comes to the specific nature of certain verbal situations. How are the wanted entailments included, or captured, within the representations of the given theory, and what are the similarities and differences between the three theories, when it comes to the situations in question?

Such entailment phenomena involve the use of verbs in specific kinds of syntactic constructions, where the sentences in question may differ from each other only slightly, or by the use of entirely different verbs altogether. We have already seen several examples of entailment phenomena in earlier chapters, for example in sections 2.3 and 3.7. Many of the upcoming examples will look at only two sentences and the entailment that occurs between them. A typical example might look at the difference between the transitive and the intransitive use of a given verb. However, some examples might also include more than two sentences, so they might vary a bit in overall complexity.

The choice of the particular verbs, and the sentences they are used in, were based strongly on each author's own examination of them. Certain verbs are more popular to discuss than others, of course, so the ones that have been examined by both Jackendoff and Parsons, as well as being represented in the VerbNet database, were the first candidates for inclusion in this analysis.

However, the choices of verbs are also based on Levin's list of prototypical verb classes, which cover a wide range of specific verbal meanings and have many different, though similar, verbs as members. These classes are of course the inspiration for VerbNet's verb classes, and they provide a fitting way to organize the entailment examples. In the end, a string of verb examples have been covered and they are neatly listed under their respective verbal classes, provided by Levin.

Note that Levin originally proposed close to fifty different prototypical verb classes, so the inclusion of only seven of them in this thesis may seem to cover much too little. However, because the focus of this analysis is on

the ways the three different representations cope with various entailment phenomena, detailed analyses of a few important verbs is more valuable than a shallow coverage of every kind of meaning. The focus is on each theory's strong and weak points, and how they capture kinds of entailments that may be shared across a wide range of different verbs. In other words, most of the verbs used in the examples could have been replaced by verbs with similar meanings, but the semantic structures would have remained practically the same.

Because this thesis is based on Jackendoff's and Parsons' most all-encompassing works, with regards to the semantic frameworks they stand behind (Jackendoff [15] and Parsons [25]), it seemed most logical, and necessary, to include verbal situations that both of them have given final lexical entries for. Therefore, most of the verbs that will be discussed cover semantic structures that were taken directly from these works, although with added specifics of the sentences in question. For instance, although Jackendoff provides final lexical entries for many verbs, the participants in those entries are unspecified. But in the coming example sentences, the participant arguments are obviously included.

However, there are occasions in certain investigations of entailment phenomena where one author does not provide a clear semantic structure for the given verb or verbs. The choice of such cases were based on the great focus they have received in the linguistic literature and on the fact that the other representation (either Jackendoff or Parsons) provides a thoughtful analysis that warrants further discussion. Still, both the conceptual structure theory and the theory of underlying events provide good enough frameworks, so that the semantic representation of a new verb is often quite easily formed. These will then be new structures that will be introduced, though they obviously conform to the rules of the given frameworks.

When it comes to VerbNet, there is much more coverage. Since the database provides a list of over 5200 different verb senses, while Jackendoff and Parsons only lists about a hundred each, it is easy to find the structure that you need for a given sentence in VerbNet. However, there might also be instances even with VerbNet where a wanted representation is not provided. In such cases, new suggestions for structures will also be proposed, though this will illustrate a weakness in the database, since it's supposed to be all-encompassing.

All of the discussions of the various entailment examples will first involve the introduction of the entailment phenomenon itself, then an examination of each theory's given structures. The different representations will be listed in the same alphabetical order as the example sentences, unless something else is specified. In the end, an overall look at the differences and similarities of the three representations will be provided, which may also discuss other important aspects, related to the given issue.

## Chapter 7

# Verbs of Contact by Impact

These kind of verbs involve the contact between participants, though differ from pure contact verbs (such as *touch*) in that the action is done with an impact, typically harming one of the participants. This chapter will examine entailments involving the contact verbs *stab* and *hit*, though other possible verbs include *strike*, *kick* and *whack*, for instance [17, p. 148].

### 7.1 Stab

This entailment example will be looking at the inclusion and exclusion of optional prepositional adjuncts, which tend to alter the meaning somewhat. The example sentences, borrowed from Parsons [25, p. 14], are all transitive, where a subject and a direct object are both realized. Though it's not included here, the intransitive version, where the direct object is removed, is also possible for the verb *stab*. However, the focus will be on the presence or non-presence of prepositional adjuncts here.

The general entailment that we would want a semantic representation to capture is the fact that a sentence that has, for instance, two optional adjuncts should entail the same sentence where either one of those adjuncts is removed, as well as the sentence where both of them are. To illustrate this, take a look at the the sentences in (78) (some of them are repeated from section 4.7), which show how the verb might be used in transitive constructions in the context of a stabbing of Caesar by Brutus.

- (78) a. *Brutus stabbed Caesar in the back with a knife.*  
b. *Brutus stabbed Caesar in the back.*  
c. *Brutus stabbed Caesar with a knife.*  
d. *Brutus stabbed Caesar.*

We see here that the core transitive sentence (78d) contains a subject (*Brutus*) and a direct object (*Caesar*), while the other sentences are variations of this sentence because of added prepositional phrases. Sentence (78a) has two such phrases added, namely *in the back* and *with a knife*, which gives additional information about where the stabbing happened on Caesar's body and what Instrument Brutus used. In contrast, sentences (78b) and

(78c) each has one of these adjuncts removed, with sentence (78b) only using *in the back* and sentence (78c) only using *with a knife*.

Sentence (78a) entails sentences (78b) and (78c) because it is both certain that the stabbing happened in the back, as in sentence (78b), and with a knife, as in sentence (78c). It also entails sentence (78d) because it is also certain that Brutus stabbed Caesar, as is the case in sentence (78d) as well. Sentence (78b), on the other hand, only entails sentence (78d), not sentence (78c), because it may be that the stabbing was done with some other Instrument than a knife. Similarly, sentence (78c) does not entail sentence (78b) because the stabbing might have happened at some other place than Caesar's back.

### 7.1.1 Jackendoff's Conceptual Structure Representation

Jackendoff has not provided a final structure for the use of the verb *stab*, so the conceptual structures given here for the example sentences are only those that are assumed to be the correct representations, based on his theory. Basically, we can imagine that a stabbing involves an Agent causing some Instrument to be at some location, which would typically be a part of some Theme. For this, the inspiration comes from Jackendoff's structure for the verb *hit* [15, p. 143].

The structures in (79) represent the four sentences. Note that structure (79a) is more complex, using the  $AFF^+$  and *BY* functions. These have been excluded from the other structures in order to make them more readable. Basically, the  $AFF^+$  function should be part of all the structures (they can be thought of as invisible), because it says that the Agent intentionally affects the Theme. The *BY* function, on the other hand, should only be part of structures (79a) and (79c), because they are the only ones that make use of an Instrument (remember that the *BY* function represents the relationship between the Instrument and the other participants).

- (79) a. [event CAUSE ([thing BRUTUS],  
[event INCH ([state BE<sub>LOC</sub> ([thing KNIFE],  
[place AT ([thing BACK PART ([thing CAESAR])])])])])])])  
AFF<sup>+</sup> ([thing BRUTUS], [thing CAESAR])  
[BY [event CAUSE ([thing BRUTUS],  
[event AFF<sup>+</sup> ([thing KNIFE],  
[thing BACK PART ([thing CAESAR])])])])])  
AFF<sup>+</sup> ([thing BRUTUS], [thing KNIFE])])]
- b. [event CAUSE ([thing BRUTUS],  
[event INCH ([state BE<sub>LOC</sub> ([thing ],  
[place AT ([thing BACK PART ([thing CAESAR])])])])])])]
- c. [event CAUSE ([thing BRUTUS],  
[event INCH ([state BE<sub>LOC</sub> ([thing KNIFE],  
[place AT ([thing PART ([thing CAESAR])])])])])])]
- d. [event CAUSE ([thing BRUTUS],  
[event INCH ([state BE<sub>LOC</sub> ([thing ],  
[place AT ([thing PART ([thing CAESAR])])])])])])]

Each of the four sentences are causing events where the Agent is *Brutus* and the Theme is *Caesar*. The immediate effect is the inchoative event resulting from the placement of some thing, which is specified as a *knife* in structures (79a) and (79c), by the use of the  $BE_{LOC}$  function. In all cases, the place in question is part of Caesar, which is represented by the use of the  $PART$  function. This location is further specified as Caesar's back in structures (79a) and (79b). The additional  $BY$  function used in the sentences mentioned above specifies that Brutus causes the affection event between the knife and Caesar's back. In this causing event, Brutus also volitionally affects the knife.

It might be hard to see how the required entailments are captured by these constructions, so the alternative flat versions might be useful as illustrations. The structures in (80) show these alternate representations. Note that the variables  $e$  and  $e_2$  stand for the causing and inchoative events, respectively, while the  $s$  variable stands for the  $BE_{LOC}$  state. Also,  $p$  is the place variable used in the  $BE_{LOC}$  function, the  $t$  variable represents the thing argument to the  $CAUSE$  function,  $t_2$  the thing argument to the  $PART$  function,  $t_3$  the thing argument to the  $BE_{LOC}$  function, and  $t_4$  the thing argument to the  $AT$  function. Finally, note that the  $AFF^+$  and  $BY$  functions have been excluded here.

- (80) a.  $[CAUSE(e) \ \& \ ARG1(e, t) \ \& \ ARG2(e, e_2) \ \& \ INCH(e_2) \ \& \ ARG1(e_2, s) \ \& \ BE_{LOC}(s) \ \& \ ARG1(s, t_3) \ \& \ ARG2(s, p) \ \& \ AT(p) \ \& \ ARG1(p, t_4) \ \& \ PART(t_4, t_2) \ \& \ BRUTUS(t) \ \& \ CAESAR(t_2) \ \& \ KNIFE(t_3) \ \& \ BACK(t_4)]$
- b.  $[CAUSE(e) \ \& \ ARG1(e, t) \ \& \ ARG2(e, e_2) \ \& \ INCH(e_2) \ \& \ ARG1(e_2, s) \ \& \ BE_{LOC}(s) \ \& \ ARG1(s, t_3) \ \& \ ARG2(s, p) \ \& \ AT(p) \ \& \ ARG1(p, t_4) \ \& \ PART(t_4, t_2) \ \& \ BRUTUS(t) \ \& \ CAESAR(t_2) \ \& \ BACK(t_4)]$
- c.  $[CAUSE(e) \ \& \ ARG1(e, t) \ \& \ ARG2(e, e_2) \ \& \ INCH(e_2) \ \& \ ARG1(e_2, s) \ \& \ BE_{LOC}(s) \ \& \ ARG1(s, t_3) \ \& \ ARG2(s, p) \ \& \ AT(p) \ \& \ ARG1(p, t_4) \ \& \ PART(t_4, t_2) \ \& \ BRUTUS(t) \ \& \ CAESAR(t_2) \ \& \ KNIFE(t_3)]$
- d.  $[CAUSE(e) \ \& \ ARG1(e, t) \ \& \ ARG2(e, e_2) \ \& \ INCH(e_2) \ \& \ ARG1(e_2, s) \ \& \ BE_{LOC}(s) \ \& \ ARG1(s, t_3) \ \& \ ARG2(s, p) \ \& \ AT(p) \ \& \ ARG1(p, t_4) \ \& \ PART(t_4, t_2) \ \& \ BRUTUS(t) \ \& \ CAESAR(t_2)]$

The only difference between the four structures is the inclusion or exclusion of the *KNIFE* and *BACK* functions, which represent the fact that the Instrument used was a knife, and that the place on Caesar's body that was stabbed was his back, respectively. Structure (80a) has both of these, structure (80b) has *BACK*, but not *KNIFE*, structure (80c) has *KNIFE*, but not *BACK*, and structure (80d) has none of them. Assuming that a formula entails another formula if it is made up of the same functions, similar to how entailments work in Parsons' theory and in VerbNet (see sections 4.7 and 5.3.4), then the correct entailments are captured by these structures (and consequently by the embedded conceptual structures). In other words, structure (80a) entails the other formulas because they consist

of the exact same functions, except that *KNIFE* and *BACK* are missing in some of them. Structures (80b) and (80c) entail structure (80d) for similar reasons.

### 7.1.2 Parsons' Underlying Event Representation

In contrast, let's analyze the way Parsons handles this situation. Because these example sentences were originally used by Parsons, the structures shown in (81) below are the exact formulas used in Parsons [25, p. 14], although they are somewhat altered in order to conform to the rules used in the final version of the theory (particularly the use of thematic role predicates, and time variables). Note that Parsons also assumes a meaning postulate, shown in (81e), which says that all stabbing events involves an Instrument [25, p. 90]. This results in the fact that there exists an unknown Instrument in formula (81d) even though the predicate is not present in the formula. Note that Parsons originally used the predicate *With*, whereas the predicate *Instrument* is used here, as mentioned in section 4.3.

- (81) a.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Stabbing}(e) \ \& \ \text{Agent}(e, \text{Brutus}) \ \& \ \text{Theme}(e, \text{Caesar}) \ \& \ \text{In}(e, \text{Back}) \ \& \ \text{Instrument}(e, \text{Knife})]$
- b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Stabbing}(e) \ \& \ \text{Agent}(e, \text{Brutus}) \ \& \ \text{Theme}(e, \text{Caesar}) \ \& \ \text{In}(e, \text{Back})]$
- c.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Stabbing}(e) \ \& \ \text{Agent}(e, \text{Brutus}) \ \& \ \text{Theme}(e, \text{Caesar}) \ \& \ \text{Instrument}(e, \text{Knife})]$
- d.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Stabbing}(e) \ \& \ \text{Agent}(e, \text{Brutus}) \ \& \ \text{Theme}(e, \text{Caesar})]$
- e. Meaning postulate:  $(\exists e) [\text{Stabbing}(e) \rightarrow (\exists x) \text{Instrument}(e, x)]$

The Parsons method is to identify the event in question as a stabbing event and appointing *Brutus* as the Agent and *Caesar* as the Theme. The additional adjunct modifiers are added as the predicates *In* and *Instrument*. In formulas (81a) and (81b) the location of the stabbing is specified to be in the *Back*, and in formulas (81a) and (81c) the Instrument is known to be a *Knife*. Formula (81d) lacks any of these adjuncts, though the existence of an Instrument is verified by the meaning postulate, as mentioned. All the events happened at some time in an interval before the present time, and they also culminated at that time.

The only difference between the four sentences concerns the explicit existence of the predicates *In* and *Instrument*. The Instrument predicate is only present in formulas (81a) and (81c) (though the meaning postulate does entail that an unknown Instrument exists in the other two), while the *In* predicate is only present in formulas (81a) and (81b). As such, the correct entailments are perfectly captured in these representations because the structures that are entailed by others contain only predicates present in those structures.

Formula (81a) has all the predicates in formula (81b), plus an Instrument predicate with a specified second argument, so formula (81a) entails

formula (81b). For similar reasons, the other entailments are also captured. Note that formula (81b) does not entail formula (81c), and vice versa, because each has a predicate that does not exist in the other. Although formulas (81b) and (81d) both have an Instrument according to the meaning postulate, no wrong entailments are present since the second argument to the predicate is unspecified, whereas it has the value *Knife* in formulas (81a) and (81c).

### 7.1.3 VerbNet's Semantic Representation

Finally, let's look at the way VerbNet represents the meaning of these sentences. The verb *stab* is a member of the verb classes *poison-42.2*, *poke-19* and *swat-18.2*, which constitute three different senses, as proposed by VerbNet. To give an illustration of how these three classes behave, the semantic structures for the sentence in (78d) are shown in (82), where structure (82a) comes from the *poison-42.2* class, structure (82b) is the *poke-19* representation and structure (82c) is the *swat-18.2* representation. Note that the *A* subscript constitutes the Agent role, *I* the Instrument, and *P* the Patient role.

- (82) a. CAUSE(Brutus<sub>A</sub>, E) & HARMED(DURING(E), Caesar<sub>P</sub>)  
 b. CAUSE(Brutus<sub>A</sub>, E) & MANNER(DURING(E), DirectedMotion, ?Instrument) & MANNER(DURING(E), Forceful, ?Instrument) & CONTACT(DURING(E), ?Instrument, Caesar<sub>P</sub>)  
 c. CAUSE(Brutus<sub>A</sub>, E) & MANNER(DURING(E), DirectedMotion, Brutus<sub>A</sub>) & NOT(CONTACT(DURING(E), Brutus<sub>A</sub>, Caesar<sub>P</sub>)) & MANNER(END(E), Forceful, Brutus<sub>A</sub>) & CONTACT(END(E), Brutus<sub>A</sub>, Caesar<sub>P</sub>))

Structures (82b) and (82c) are more similar to each other than structure (82a), as structure (82a) uses a very simple predicate called *HARMED*, which we can assume only signifies that the Patient is harmed during the event. The other two give more details about the manner of the event and about the contact between Patient, Instrument and Agent. The difference between structures (82b) and (82c) is that in structure (82b) the focus is on an unknown Instrument (which is known in the sentences in (78a)) and (78c), while in structure (82c) the focus is on the Agent.

In other words, the manner during and at the end the event, which is a *DirectedMotion* and *Forceful*, respectively, is appointed to the Instrument in structure (82b) and to the Agent in structure (82c). Also, both structures signify a contact between one participant and the Patient at the end of the event, but in structure (82b) this participant is the Instrument, while it is the Agent in structure (82c). In addition, structure (82c) says that there is no contact between the Agent and the Patient during the event. Structure (82b) does not say anything about this. Given that it is assumed that there is an unknown Instrument in sentence (78d) (if Brutus stabs Caesar it is assumed that Brutus uses some form of instrument, not himself) we can assume that the most relevant structure here is the one in structure (82b).

So, this section will be concentrating on the *poke-19* class when discussing the VerbNet representations of the sentences in question.

Note, however, that *poke-19* lacks two relevant thematic roles that are present in *swat-18.2*, namely a Location and a Result role. The Location role has, in the *swat-18.2* class, a selectional restriction which says that it must be concrete, and in the one frame where it is used it is also restricted to being a body part. However, in that frame, which would represent the sentence in (78b), the Location role is not used in the semantic structure, which seems a bit strange. The Result role might have been realized with a phrase such as *to death*, as in *Brutus stabbed Caesar to death*. Because these two roles are not present in *poke-19*, the section will dedicate some discussion to the semantic structures in *swat-18.2* later, after going through the *poke-19* structures.

Finally, note also that the Instrument role has an additional selectional restriction in the *poke-19* class which is not present in *swat.18.2*. This restriction says that the Instrument must be *pointy* (the other class instead says that the Instrument must be *solid*), something which feels right given the sentences under discussion.

The semantic structures (83), coming from the *poke-19*, represent the four sentences. Because the class does not consider the location where the Patient is stabbed, there is only a difference between the structures that have an Instrument and those that don't.

- (83) a. CAUSE(Brutus<sub>A</sub>, E) &  
MANNER(DURING(E), DirectedMotion, Knife<sub>I</sub>) &  
MANNER(END(E), Forceful, Knife<sub>I</sub>) &  
CONTACT(END(E), Knife<sub>I</sub>, Caesar<sub>P</sub>) &  
USE(DURING(E), Brutus<sub>A</sub>, Knife<sub>I</sub>)
- b. CAUSE(Brutus<sub>A</sub>, E) &  
MANNER(DURING(E), DirectedMotion, ?Instrument) &  
MANNER(END(E), Forceful, ?Instrument) &  
CONTACT(END(E), ?Instrument, Caesar<sub>P</sub>)
- c. CAUSE(Brutus<sub>A</sub>, E) &  
MANNER(DURING(E), DirectedMotion, Knife<sub>I</sub>) &  
MANNER(END(E), Forceful, Knife<sub>I</sub>) &  
CONTACT(END(E), Knife<sub>I</sub>, Caesar<sub>P</sub>) &  
USE(DURING(E), Brutus<sub>A</sub>, Knife<sub>I</sub>)
- d. CAUSE(Brutus<sub>A</sub>, E) &  
MANNER(DURING(E), DirectedMotion, ?Instrument) &  
MANNER(END(E), Forceful, ?Instrument) &  
CONTACT(END(E), ?Instrument, Caesar<sub>P</sub>)

While the fact that structure (83a) should entail the other sentences, and the fact that structures (83b) and (83c) should both entail structure (83d) are captured here, there is an error of entailment between structures (83b) and (83c). Structure (83c) incorrectly entails structure (83b) because a mapping of the location is missing in structure (83b). In other words, structure (83b) is almost identical to structure (83c), except for the specified Instrument,



but there is no representation of a Location in structure (83b) which would make it distinct from the other. A solution to this incorrect entailment might be to add a Location role to the given class, and use it within the semantic representations where it is realized.

However, because of these wrong entailments, let's consider the *swat-18.2* class instead, since it does include a Location role. The semantic structures are shown in (84) below.

- (84) a. CAUSE(Brutus<sub>A</sub>, E) &  
 MANNER(DURING(E), DirectedMotion, Knife<sub>I</sub>) &  
 NOT(CONTACT(DURING(E), Knife<sub>I</sub>, Caesar<sub>P</sub>)) &  
 MANNER(END(E), Forceful, Knife<sub>I</sub>) &  
 CONTACT(END(E), Knife<sub>I</sub>, Caesar<sub>P</sub>) &  
 USE(DURING(E), Brutus<sub>A</sub>, Knife<sub>I</sub>)
- b. CAUSE(Brutus<sub>A</sub>, E) &  
 MANNER(DURING(E), DirectedMotion, Brutus<sub>A</sub>) &  
 NOT(CONTACT(DURING(E), Brutus<sub>A</sub>, Caesar<sub>P</sub>)) &  
 MANNER(END(E), Forceful, Brutus<sub>A</sub>) &  
 CONTACT(END(E), Brutus<sub>A</sub>, Caesar<sub>P</sub>))
- c. CAUSE(Brutus<sub>A</sub>, E) &  
 MANNER(DURING(E), DirectedMotion, Knife<sub>I</sub>) &  
 NOT(CONTACT(DURING(E), Knife<sub>I</sub>, Caesar<sub>P</sub>)) &  
 MANNER(END(E), Forceful, Knife<sub>I</sub>) &  
 CONTACT(END(E), Knife<sub>I</sub>, Caesar<sub>P</sub>) &  
 USE(DURING(E), Brutus<sub>A</sub>, Knife<sub>I</sub>)
- d. CAUSE(Brutus<sub>A</sub>, E) &  
 MANNER(DURING(E), DirectedMotion, Brutus<sub>A</sub>) &  
 NOT(CONTACT(DURING(E), Brutus<sub>A</sub>, Caesar<sub>P</sub>)) &  
 MANNER(END(E), Forceful, Brutus<sub>A</sub>) &  
 CONTACT(END(E), Brutus<sub>A</sub>, Caesar<sub>P</sub>))

As mentioned, when the Location role is a part of the syntactic structure, as is the case for the frames relevant to structures (84a) and (84b), it does not take part in the semantic structure. Because of this, notice that structures (84a) and (84c) are exactly the same, which is also the case for structures (84b) and (84d). This then results in the same entailment errors we saw with the *poke-19* structures. This leads to the conclusion that the wanted entailments are not captured in the current form of VerbNet.

On a side note, it is interesting how there are two possible ways to encode the location in VerbNet. One is by using a predicate called *LOCATION* and the other is to make use of the thematic role *Location*. In the class *spray-9.7*, for instance, the semantic predicate *LOCATION* is used (there is no Location role, although there is a Destination role, which is a sub-role of Location [24, p. 18]). The predicate denotes the whereabouts of the Theme in accordance with the Destination role. For example, in a sentence like *John loaded the truck with hay*, the location of the hay at the end of the event is represented by the predicate *LOCATION(END(E),*

*Hay<sub>T</sub>, Truck<sub>D</sub>*), where the *T* subscript represents the Theme, and the *D* the Destination.

When the location of an event is an adjunct that is not obligatory, realized as a prepositional phrase in a sentence like *Brutus stabbed Caesar in Oslo*, the information is not a part of the semantic structure, because the phrase does not alter the core meaning of the event. The Location role is, in other words, only part of a structure in the class *swat-18.2* if the participant occupying the Location role is both concrete and a body-part, and if the syntactic form of the sentence is *NP V NP PP.Location*. The question is then, which was already noted, why the semantic predicate *LOCATION*, or any other predicate that may represent the location information, is not used in the semantics of that frame, as it is in the class *spray-9.7*, for instance.

#### 7.1.4 Notable Differences and Similarities

Although both Jackendoff and VerbNet treat the intended meaning as a causative, using the function *CAUSE* and the predicate *CAUSE*, respectively, Parsons' formulas do not directly convey any such meaning. In other words, rather than including the verb *stab* in his list of causative-inchoatives [25, p. 268], he includes it in his list of transitive verbs [25, p. 266]. Still, *Brutus* is applied the role of Agent, a role that is defined as the participant that causes the event or affects the Theme in some way. So, while the available predicate *CAUSE* is not used in Parsons' formula, the fact that it is Brutus that affects Caesar is still captured (remember that *CAUSE* in Parsons' theory is used to represent a relationship between events, not between participants and events).

In relation to this, Jackendoff makes use of two events for these sentences, namely the causative event where Brutus is the causer, and the effect event, which is an inchoative. The other two representations, on the other hand, only refer to one event here. However, because the second event in Jackendoff's structure is embedded inside the *CAUSE* event, it might be appropriate to say that there is only one event overall, and that the second event is only a smaller part of the bigger event. If we adopt this view then all three representations refer to only one event here, although with Jackendoff's structures another event is part of the structure of the overall event.

The only information that Parsons provides regarding the nature of the stabbing is in the *Stabbing* predicate itself. In other words, there is no mention of what characterizes a stabbing and what makes it different from other, similar verbs. What would be the difference between the formula for a sentence like *Brutus sliced Caesar with a knife* and (81c), for instance, other than the predicates *Slicing* and *Stabbing*? And how would the fact that these are events representing very similar verbs, which both involve moving some object to cause some destruction, be captured in Parsons' formula? This seems to be an overall limitation of the underlying event theory; the fact that there is too little detail in the predicates describing particular events.

VerbNet, and Jackendoff to a lesser degree, describes in more detail

what actually goes on within the various stages of the event. When the knife is present in a sentence, both the fact that it moves in a directed motion during the event, and is forceful at the end of it, is captured in VerbNet. The fact that there is no contact between the knife and Caesar during the event, but there is at the end, is also represented. Such information is missing in Parsons' and Jackendoff's structures, although Jackendoff does represent the fact that the knife ends up at a place which is part of Caesar (either his back or something else), with the use of the locative *BE* state function.

This brings us to another issue. The fact that it is Caesar's back that is stabbed is clearly captured in Jackendoff's structures, by the use of the *PART* function. In Parsons formula, however, there is no predicate that links the two participants *Back* and *Caesar* together. A possible solution to this might be to change the name of the *Back* to *Caesar's back*, but this change would really only affect the name of the participant, not the meaning behind it. So, another solution might be to introduce a predicate that links the participants, making the back a part of Caesar (perhaps with a predicate called *Part* that would take two linked entities). In the VerbNet structures the fact that the location is Caesar's back isn't represented at all (as mentioned earlier, with the issue of the location not being part of the semantic structure).

The structures that best represent the tense of the sentences are clearly Parsons' formulas. Nowhere in Jackendoff's or VerbNet's structures is it mentioned that the event took place in the past. This shows that there is a stronger relationship between syntax and semantics in Parsons' framework than in the other two, which focus more on the core meaning of the sentences. So, while there is a difference in the underlying event theory between a sentence in the present tense and one in the past tense, there is no focus on such a difference in the other two frameworks.

## 7.2 Hit

The verb *hit* can be used in conjunction with an Instrument, which is typically used by an Agent in order to affect some Theme. A sentence involving all of these roles can have many different surface realizations, a typical one being the use of a *with*-phrase to include the Instrument. In that case, the Instrument appears as an argument to the prepositional phrase, while the Theme typically appears in the direct object position. However, the reverse positioning occurs when the preposition is *against* instead of *with*; the Instrument is then the direct object and the Theme acts in the prepositional phrase as an argument. Effectively, the two different surface realizations can refer to the same event, but from different perspectives. Additionally, *hit* can be used in a purely transitive sentence without any prepositional adjuncts. In that case, both the Agent and the Instrument can act as the subject [17, p. 67].

Consider the examples in (85) below, where sentence (85a) shows the use of a *with*-phrase, sentence (85b) the use of an *against*-phrase, sentence

(85c) the use of the pure transitive variant with the Agent (*John*) as the subject, and sentence (85d) with the Instrument (*the hammer*) as the subject.

- (85) a. *John hit the nail with the hammer.*  
b. *John hit the hammer against the nail.*  
c. *John hit the nail.*  
d. *The hammer hit the nail.*

Here, the two sentences in (85a) and (85b) both entail the sentence in (85d), because in both cases it is apparent that it is the hammer that comes into contact with the nail, and not John himself. If we assume that sentences (85a) and (85b) refer to the same event, then they also should entail one another. In sentence (85c), on the other hand, there is no mention of an Instrument being used, so the logical conclusion is that it is John who directly hit the nail, by not using any Instrument at all. However, sentence (85c) could perhaps be understood as there being an obligatory unknown Instrument that is just not mentioned, similar to the verb *stab*, discussed in section 7.1. The Instrument could then just be a part of the Agent's body, like his fist, for instance. But if we understand sentences (85a) and (85b) as there being a contact between the hammer and the nail, with John as the causer behind it, and sentence (85c) as there being a direct contact between some body part of John and the nail, then logically the two sentences in (85a) and (85b) do not entail sentence (85c), though they do, as mentioned, entail the sentence in (85d).

### 7.2.1 Jackendoff's Conceptual Structure Representation

Jackendoff emphasizes that the verb *hit* can have two different analyses, namely as a motion verb or as an inchoative [15, p. 109-110, 142-145]. The structures in (86) below show how the pure transitive sentence in (85d) would look like using these two alternations, one using the *GO* event function and the other using the *INCH* event function.

- (86) a. [event GO<sup>+C</sup> ([thing HAMMER],  
[path TO ([place AT ([thing NAIL]))]])]  
b. [event INCH ([state BE<sup>+C</sup> ([thing HAMMER],  
[place AT ([thing NAIL]))]])]

Remember that the +C superscript constitutes that the function involves contact between arguments, here the contact between the hammer and the nail. Jackendoff proposes that the structure in (86b) is the correct one for the sentences in question because the version in (86a) would permit continuous contact with the nail and not a sudden one. The fact that the motion of the hammer should culminate when coming into contact with the nail is not captured directly with structure (86a), but it is with structure (86b) because it describes a state coming about. Structure (86a) could, for instance, signify a meaning similar to the sentence *The hammer stroked the nail*, which is instead an event with continuous movement.

In other words, the two structures illustrate two types of contact meanings, namely *moving* contact, in structure (86a), and *impact* contact, in structure (86b). A third contact meaning type called *pure* contact is also possible, which occurs with verbs like *touch* and *contact*. This type is represented simply as a *BE* state, without a surrounding *INCH* event (as in structure (86b)).

So, the inchoative variant should be used for the verb in this case because the intended meaning (an impact) follows automatically from it. The fact that it is *John* who acts as the causing Agent in the first three sentences is captured by the use of the *CAUSE* function. In other words, the sentence in (85b), for instance, can be paraphrased as *John caused the hammer to hit against the nail*.

The conceptual structures for the four sentences in question are shown in (87) below. Notice that only structures (87a) and (87b) include the *BY* function, since only those two mention an Instrument. Also, it should be noted that although the three structures in (87a), (87c) and (87d) were specifically proposed by Jackendoff [15, p. 143], there is some uncertainty as to what the structure in (87b) should use as its place function. For now, let's that a possible solution is to use the place function *AGAINST*, since that is what is used in the sentence.

- (87) a. [event CAUSE ([thing JOHN],  
[event INCH ([state BE<sup>+C</sup> ([thing HAMMER],  
[place AT ([thing NAIL])])])])]  
AFF<sup>+</sup> ([thing JOHN], [thing NAIL])  
[BY [event CAUSE ([thing JOHN],  
[AFF<sup>-</sup> ([thing HAMMER], [thing NAIL])]  
AFF<sup>+</sup> ([thing JOHN], [thing HAMMER])]
- b. [event CAUSE ([thing JOHN],  
[event INCH ([state BE<sup>+C</sup> ([thing HAMMER],  
[place AGAINST ([thing NAIL])])])])]  
AFF<sup>+</sup> ([thing JOHN], [thing NAIL])  
[BY [event CAUSE ([thing JOHN],  
[AFF<sup>-</sup> ([thing HAMMER], [thing NAIL])]  
AFF<sup>+</sup> ([thing JOHN], [thing HAMMER])]
- c. [event INCH ([state BE<sup>+C</sup> ([thing JOHN],  
[place AT ([thing NAIL])])])]  
AFF<sup>+</sup> ([thing JOHN], [thing NAIL])]
- d. [event INCH ([state BE<sup>+C</sup> ([thing HAMMER],  
[place AT ([thing NAIL])])])]  
AFF<sup>-</sup> ([thing HAMMER], [thing NAIL])]

Structures (87a) and (87b) are pretty much the same, except for having two different place functions (*AT* and *AGAINST*). In both cases, we have a *CAUSE* event where the Agent, *John*, causes an *INCH* event that takes a *BE* state as an argument. This state involves contact, represented by the +C superscript. In the state, the *Hammer* is placed either at or against the

*NAIL*. Also in both cases, John volitionally affects the nail, and he does this by (using the *BY* function) causing the hammer to involuntarily affect the nail. In this other causing event, John also volitionally affects the hammer.

The structures in (87c) and (87d) are a bit simpler since they lack the *BY* function. They are both *INCH* events and their structures are basically the same as the *INCH* events in structures (87a) and (87b) (plus the *AFF* function), except for the different things involved. In structure (87c) it is John that is placed at the nail, whereas in structure (87d) it is instead the hammer that is placed there. In both of them, the thing in question affects the nail, but while this is done volitionally in structure (87c), it is not in structure (87d) (since a hammer can't really act on its own intentionally).

Because the *INCH* function in structure (87d) is exactly the same as in structure (87a) and the *AFF* function that is present in structure (87d) is also present inside the *BY* function in structure (87a), it is apparent that structure (87a) entails structure (87d). This is also the case in the relationship between structures (87b) and (87d), except for the fact that the *AGAINST* place function is used instead. However, if we assume that using the *AT* function conveys the same meaning; that the structure for the sentence in (85b) is the same as the one in (87a), then structure (87b) also entails structure (87d). Still, this does illustrate that it can be hard to know which function one should use based on the sentence given, and how such different functions result in entailments not being captured. Additionally, one can see that structures (87a) and (87b) do not entail structure (87c), since structure (87c) is not part of any of the other two. So, these representations correctly capture the wanted entailments.

## 7.2.2 Parsons' Underlying Event Representation

Following the template provided by Parsons [25, p. 260], which suggests a formula for the transitive verb *hit* [25, p. 266], the resulting representations for the given sentences are shown in (88). Note that the template says that when *hit* is involved, the subject can be either an Agent or a Performer, where the former acts in formulas (88a), (88b) and (88c), and the latter acts in formula (88d). Remember that a Performer is something that has no immediate agency and is not directly controlled by anything, which would be the case when *the hammer* is the subject.

- (88) a.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Hitting}(e) \ \& \ \text{Agent}(e, \text{John}) \ \& \ \text{Theme}(e, \text{Nail}) \ \& \ \text{Instrument}(e, \text{Hammer})]$   
 b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Hitting}(e) \ \& \ \text{Agent}(e, \text{John}) \ \& \ \text{Theme}(e, \text{Hammer}) \ \& \ \text{Against}(e, \text{Nail})]$   
 c.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Hitting}(e) \ \& \ \text{Agent}(e, \text{John}) \ \& \ \text{Theme}(e, \text{Nail})]$   
 d.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Hitting}(e) \ \& \ \text{Performer}(e, \text{Hammer}) \ \& \ \text{Theme}(e, \text{Nail})]$

All of these formulas involve a *Hitting* event that happened in the past, and they all refer to some Theme taking part in that event. However,

assuming that the two prepositional phrases in sentences (85a) and (85b) result in two adverbial predicates called *With* and *Against*, respectively, in Parsons' theory (remember the use of the predicate *Instrument* instead of *With*), the formulas in (88a) and (88b) are quite different from each other. Also assuming that the Theme is directly mapped from the direct object in the sentence, the two formulas differ when it comes to what this Theme is and in their use of adverbial predicates.

Formula (88c) is basically the same as the one in (88a), except without the *Instrument* predicate, while the formula in (88d) is very much the same as the one in (88c), using, for instance, the same Theme. However, while formula (88c) appoints *John* as the Agent, in formula (88d) there is no Agent but rather a Performer.

If we assume that the formulas are the correct ones for this framework, none of the wanted entailments are captured. Neither formula (88a) nor formula (88b) entail formula (88d) because the latter uses a predicate that is not present in the other two (the *Performer* predicate). Formulas (88a) and (88b) are a little too different from each other, considering the fact that the sentences they represent can be used to refer to the same event, according to Fillmore [9, p. 75]. In addition, a seemingly incorrect entailment is present between formulas (88a) and (88c), since both are the same, except for the present *Instrument* predicate in formula (88a). However, this entailment might in fact be correct after all, since John does indeed hit the nail in both cases (though the Instrument used is known in formula (88a) but unknown in formula (88c)).

For the correct entailments to be captured, the participants should have the same thematic roles in all the sentences. *John* should be the Agent, *the nail* should be the Theme and *the hammer* should be the Instrument. However, it is unclear how this should come about when the sentences are mapped to the formulas. How can it be known that *the nail* is the Theme when used in an *against*-phrase, and that *the hammer* can also be an Instrument used by some unknown Agent in formula (88d)? A main reason for why the wanted entailments are not captured is the fact that *the hammer* changes its thematic role to Performer when it acts as the subject. If we said that *the hammer* is either a Performer in formula (88a), or an Instrument in formula (88d), then the correct entailments would have been captured. But in the current framework, the fact that one of those options should be the case is very much unclear.

### 7.2.3 VerbNet's Semantic Representation

The verb *hit* is a member in six classes in VerbNet, namely *bump-18.4*, *contiguous\_location-47.8*, *hit-18.1-1*, *hurt-40.8.3*, *reach-51.8* and *throw-17.1-1*. However, let's assume the correct class is *hit-18.1-1*, both because *hit* is part of its name, and because it is the only class that permits the syntactic construction used in (85b), where *against* is used in conjunction with an Agent.

In addition, the class *contiguous\_location-47.8* involves a Theme and a Co-Theme and no Agent, which does not apply to our sentences. *Hurt-*

40.8.3 requires its Patient to be either a body part or reflexive, meaning that someone or John himself would need to be hurt, which also does not apply here. Finally, *reach-51.8* and *throw-17.1-1* represent two quite different senses all together. However, *bump-18.4* does provide a suitable frame for the sentence in (85b), but without the Agent (resulting in the sentence *The hammer hit against the nail*), so this will be considered briefly at the end of this section.

Using the frames available in the class *hit-18.1-1* the semantic structures for the sentences are shown in (89) below, where *A* is the Agent, *P* is the Patient and *I* is the Instrument.

- (89) a. CAUSE(*John<sub>A</sub>*, *E*) &  
 MANNER(DURING(*E*), DirectedMotion, Hammer<sub>*I*</sub>) &  
 NOT(CONTACT(DURING(*E*), Hammer<sub>*I*</sub>, Nail<sub>*P*</sub>)) &  
 MANNER(END(*E*), Forceful, Hammer<sub>*I*</sub>) &  
 CONTACT(END(*E*), Hammer<sub>*I*</sub>, Nail<sub>*P*</sub>) &  
 USE(DURING(*E*), John<sub>*A*</sub>, Hammer<sub>*I*</sub>)
- b. CAUSE(*John<sub>A</sub>*, *E*) &  
 MANNER(DURING(*E*), DirectedMotion, Hammer<sub>*I*</sub>) &  
 NOT(CONTACT(DURING(*E*), Hammer<sub>*I*</sub>, Nail<sub>*P*</sub>)) &  
 MANNER(END(*E*), Forceful, Hammer<sub>*I*</sub>) &  
 CONTACT(END(*E*), Hammer<sub>*I*</sub>, Nail<sub>*P*</sub>) &  
 USE(DURING(*E*), John<sub>*A*</sub>, Hammer<sub>*I*</sub>)
- c. CAUSE(*John<sub>A</sub>*, *E*) &  
 MANNER(DURING(*E*), DirectedMotion, John<sub>*A*</sub>) &  
 NOT(CONTACT(DURING(*E*), John<sub>*A*</sub>, Nail<sub>*P*</sub>)) &  
 MANNER(END(*E*), Forceful, John<sub>*A*</sub>) &  
 CONTACT(END(*E*), John<sub>*A*</sub>, Nail<sub>*P*</sub>)
- d. MANNER(DURING(*E*), DirectedMotion, Hammer<sub>*I*</sub>) &  
 NOT(CONTACT(DURING(*E*), Hammer<sub>*I*</sub>, Nail<sub>*P*</sub>)) &  
 MANNER(END(*E*), Forceful, Hammer<sub>*I*</sub>) &  
 CONTACT(END(*E*), Hammer<sub>*I*</sub>, Nail<sub>*P*</sub>)

The structures in (89a) and (89b) are exactly the same, even though their syntactic structures differ somewhat. In both, *John* (acting as the Agent) causes the event to happen and he also uses *the hammer* (acting as the Instrument) during the event. The manner of which the hammer behaves is that of directed motion during the event, but forceful at the end of it. Also, there is no contact between the hammer and the nail during the event, but there is at the end. Because of these same structures, the fact that they entail one another is captured.

The structure in (89c) is similar to the two structures above it, except that the focus is on the Agent and not the Instrument when it comes to manner and contact. Instead of *the hammer* being the argument to the *MANNER* predicates, *John* is used. Also, instead of *the hammer* and *the nail* being the two arguments to the *CONTACT* predicates, *John* and *the nail* are used. Because of these differences, the suggested fact that structures (89a)



and (89b) should not entail structure (89c) is captured. Also, structure (89d) consists of the same semantic predicates as in structures (89a) and (89b), though it lacks the *CAUSE* predicate. So, structures (89a) and (89b) both entail structure (89d), which is correct.

The semantic structure provided by the class *bump-18.4* for the alternative sentence *The hammer hit against the nail*, which should be entailed by the sentences in (85a) and (85b), is shown in structure (90) below. As can be seen, it is exactly the same as the one in structure (89d) except that the roles have been changed. The *hammer* is the Theme instead of the Instrument, while the *nail* is the Location and not the Patient. Because of these differences there is no entailment relation present between the frames in *hit-18.1-1* and this frame from *bump-18.4*, even though the two meanings here are very much related. Had the names of the roles been excluded, however, there would have been a relation, since the same participants are placed in the same argument positions.

- (90) MANNER(DURING(E), DirectedMotion, Hammer<sub>THEME</sub>) &  
 NOT(CONTACT(DURING(E), Hammer<sub>THEME</sub>, Nail<sub>LOCATION</sub>)) &  
 MANNER(END(E), Forceful, Hammer<sub>THEME</sub>) &  
 CONTACT(END(E), Hammer<sub>THEME</sub>, Nail<sub>LOCATION</sub>)

## 7.2.4 Notable Differences and Similarities

Although John is appointed as the Agent in all three representations for the sentences in (85a) and (85b), the role he has in sentence (85c) is different in its given conceptual structure (he is still the Agent in the structures (87a) and (87b)). In structure (87c), John appears inside the *BE<sub>LOC</sub>* function, so he then occupies the role of Theme instead of Agent.

The hammer, on the other hand, is always the Theme in both Jackendoff's structures and the VerbNet structures, though there are significant changes going on in the underlying event formulas. In formula (88a), the hammer is the Instrument, but in formula (88b) it is the Theme, and in formula (88d) it is the Performer. This role change between formulas seems quite unnatural, since these events are very closely related (the hammer does pretty much the same thing in all).

The nail acts as an argument to a place function in Jackendoff's structures, making it the Location or Goal whenever it appears in structure (87). In VerbNet, the nail is always appointed the role of Patient. However, while this participant is said to be the Theme in formulas (88a), (88c) and (88d), it instead acts as an argument to the *Against* predicate in formula (88b).

So, VerbNet is the one framework that is most consistent with its role labeling here, appointing the same roles for each participant in all structures, while the other two representations juggle between different roles for some of the participants. This depends, of course, on which class is chosen for the given sentences, as this choice may lead to different thematic role labels. This was seen in the alternative structure that *bump-18.4* provides, which appoints rather different roles to the participants.

So, while it's typically quite clear what roles the arguments in a sentence should have in Jackendoff's and Parsons' frameworks, this mapping can prove to be much harder in VerbNet, since it provides a lot more possibilities. In addition, VerbNet by itself does not know just from the sentence what the underlying meaning is supposed to be, so a clear, one-to-one choice of the right semantic structure is not possible in certain cases.

Jackendoff's conceptual structures represent the idea that either John or the hammer is placed at or against the nail, and also how the participants affect each other (either voluntarily or not). The latter information is missing in VerbNet, though those structures do have even more detailed descriptions of the event. The underlying event formulas lack any detailed descriptions of what actually happens in a *Hitting* event. This is similar to what was talked about in section 7.1.4, so this section won't go deeper into it here.

Both the structures provided by Jackendoff and VerbNet perfectly capture the wanted entailments here, most importantly the fact that sentences (85a) and (85b) should not entail sentence (85c), but only sentence (85d). The underlying event formulas, on the hand, does not capture any entailments, basically because of the different predicates used and the fact that the hammer is appointed different roles in all the structures.

## Chapter 8

# Verbs of Killing

For these verbs a participant goes from being alive to being dead, either because of some direct cause (like a killer performing a murder) or from something unknown. The coming section will be looking at the related verbs *kill* and *die*, though similar verbs, such as *murder* and *slay* are also verbs of killing. Note, however, that Levin lists *die* under a verb class involving disappearance (Verbs of Appearance, Disappearance, and Occurrence), but it's inclusion here is all because of its apparent relationship with *die* [17, p. 230,258].

### 8.1 Kill/Die

The relationship between the verbs *kill* and *die* have been briefly touched upon before, for instance in section 2.3. Though they do not share the same lemma they can still be used to describe the same event instance, only through different perspectives. For *kill* the subject (in an active sentence) is the Agent causing the death of some Theme, while for *die* the subject is instead the Theme, where the mention of an Agent is typically lacking. These different forms are also related to the given state of the Theme.

The three sentences in (91) show some uses of these two verbs, how they might be related to each other and to the adjective *dead*. Note that sentence (91a) is a causative sentence, where *John* is the Agent, sentence (91b) is an inchoative sentence, where *Bill* is the Theme (he is also the Theme in sentence (91a)), and sentence (91c) is a stative sentence describing the status of a Theme (again *Bill*).

- (91) a. *John killed Bill.*  
b. *Bill died.*  
c. *Bill is dead.*

Sentences (91a) and (91b) basically describe the same event (if we assume that *Bill* died because of *John* in sentence (91b)), which results in the state of *Bill* being dead (sentence (91c)). Logically, sentence (91a) should entail sentence (91b) because if someone is killed then that someone also dies. However, the reverse statement may not be true; if someone dies it is

not certain that they were killed by anyone. Both sentences (91a) and (91b) should also entail the state in sentence (91c), because if someone dies then that person will always be in the state of being dead following the dying event.

### 8.1.1 Jackendoff's Conceptual Structure Representation

Assuming that the conceptual structures for these sentences would involve the state of *Bill*, and the use of the functions *CAUSE* and *INCH*, the structures in (92) show how Jackendoff's theory might represent them. Note that these exact structures were not directly proposed by Jackendoff, but are rather suggestions that conform well with his given framework.

- (92) a. [event CAUSE ([thing JOHN],  
[event INCH ([state BE<sub>IDENT</sub> ([thing BILL],  
[place AT ([property DEAD])])])])]  
b. [event INCH ([state BE<sub>IDENT</sub> ([thing BILL],  
[place AT ([property DEAD])])])]  
c. [state BE<sub>IDENT</sub> ([thing BILL],  
[place AT ([property DEAD])])]

To summarize, structure (92a) involves a *CAUSE* event where *John* is the cause and an *INCH* event is the effect. This *INCH* event is the same as the event in structure (92b) which describes the coming about (the inchoative) of the state of *Bill* having the property of being dead. This state is again the same as the state in structure (92c). The entailment is captured perfectly by the fact that each entailed structure is part of another structure that leads to it. Structure (92b) is entailed from structure (92a) because it is itself an argument to the *CAUSE* function. In addition, structure (92c) is also entailed from structure (92b) (and consequently structure (92a) as well) because it is an argument to the *INCH* function.

Note that Jackendoff also proposes the rather simple function *DIE* for a sentence like the one in (91b) [15, p. 299]. The structure would then be [event *DIE*([thing *BILL*)]], though let's assume the structures in (92) for this example.

### 8.1.2 Parsons' Underlying Event Representation

Next, let's look at Parsons' formulas. Following his logical form for causative-inchoative sentences, the resulting formulas for these three sentences are shown in (93) below [25, p. 268,277].

- (93) a.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Agent}(e, \text{John}) \ \& \ (\exists e_2)(\exists t_2) [t_2 \in I \ \& \ \text{BECOME\#}(\text{Dead})(e_2) \ \& \ \text{Theme}(e_2, \text{Bill}) \ \& \ \text{Cul}(e_2, t_2) \ \& \ \text{CAUSE}(e, e_2)]]$   
b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{BECOME\#}(\text{Dead})(e) \ \& \ \text{Theme}(e, \text{Bill}) \ \& \ \text{Cul}(e, t)]$   
c.  $(\exists s)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Being-dead}(s) \ \& \ \text{Theme}(s, \text{Bill}) \ \& \ \text{Hold}(s, t) \ \& \ \neg(\exists t_2) [t_2 < t \ \& \ \text{Hold}(s, t_2)]]]$

Both formulas (93a) and (93b) are very much alike, except that formula (93a) has a surrounding structure which describes a causing event ( $e$ ), that takes place in the same interval as the other event ( $e_2$ ). The added *CAUSE* predicate constitutes the fact that  $e$  causes  $e_2$ . The event  $e$  in (93b) is then the exact same event as  $e_2$  in formula (93a), except without the *CAUSE* predicate. For instance, each has *Bill* as its Theme. Similar to the use of the *CAUSE* and *INCH* functions in the conceptual structures in (92), the *CAUSE* predicate and the *BECOME#* function represent the causative and the inchoative sentences, respectively.

Because of the embedded event in formula (93a) it is apparent that it entails formula (93b), which is what we want to be true. The entailment between formulas (93b) and (93c) is further established because of the given meaning postulate that Parsons proposes for inchoative sentences. As mentioned in section 4.5, a meaning postulate for a dying event would look like the one in (94) below, which says that if there is a becoming-dead event with  $X$  as its Theme, and it culminates, then there exists a state of being dead which holds, has the same  $X$  as its Theme, and did not previously hold. Note that the point in time,  $t$ , is included here, which was not present in Parsons' original structure.

- (94)  $\text{BECOME\#(Dead)}(e) \ \& \ \text{Cul}(e, t) \ \& \ \text{Theme}(e, X) \rightarrow$   
 $(\exists s) [\text{Being-Dead}(s) \ \& \ \text{Hold}(s, t) \ \& \ \text{Theme}(s, X) \ \& \neg \text{PREVIOUSLY}(\text{Hold}(s, t))]$

Because formula (93b) has all the prerequisite conditions that appear at the left side of the arrow in the meaning postulate in (94), the fact that a being-dead state exists is true. This state is exactly the same as the state in formula (93c). Note that the last part of the formula, which refers to the non-existence of a time point that holds before the current time, means exactly the same as  $\neg \text{PREVIOUSLY}(\text{Hold}(s, t))$ . So, similar to the Jackendoff approach, all the required entailments are captured by these formulas, plus the additional meaning postulate.

### 8.1.3 VerbNet's Semantic Representation

In VerbNet, the verb *kill* appears as a member in the classes *murder-42.1-1* and *subjugate-42.3*, though let's assume the former class is the correct for the given sense here, because it involves the semantic predicate *ALIVE*. The semantic structures in that other class only say that Bill becomes subjugated during the event (using the predicate *SUBJUGATED*), and the event is caused by Brutus. The verb *die*, on the other hand, only appears in one class, namely *disappearance-48.2*. Using the frames available in these classes, the resulting semantic structures are shown in (95). Note again that  $A$  stands for the Agent and  $P$  stands for the Patient.

- (95) a.  $\text{CAUSE}(\text{John}_A, E) \ \& \ \text{ALIVE}(\text{START}(E), \text{Bill}_P) \ \& \neg \text{ALIVE}(\text{RESULT}(E), \text{Bill}_P)$   
 b.  $\text{DISAPPEAR}(\text{DURING}(E), \text{Bill}_P)$

c. NOT(ALIVE( $E$ , Bill <sub>$P$</sub> ))

The structure in (95a) seems pretty logical. At the start of the event the Patient (*Bill*) is alive, but as a result of the event he is not alive. The cause of the event is the Agent (*John*). Structure (95b), on the other hand, uses an entirely different semantic predicate, namely *DISAPPEAR*, which seems to give a very different meaning to the event. Other members in the class include *vanish* and *disappear*, giving the impression that the meaning is not referring to losing one's life, but rather to someone's disappearance. *Die* points to two different senses in WordNet, namely the typical sense involving the "loss of all bodily attributes and functions necessary to sustain life", and disappearance; of "coming to an end" [18].

A more intuitive approach to the verb *die* would probably have been the use of the same predicates as in structure (95a), except without the *CAUSE* predicate. But because VerbNet does not offer any such alternatives, it seems appropriate to conclude that the wanted entailment is not captured in the current state of the database.

However, the fact that structure (95a) entails structure (95c) is captured, since the *ALIVE* predicate refers to the state of the given participant (which is Bill in this case). If the state of Bill not being alive is the resultant state of the event in structure (95a), then the structure in (95c) is entailed if we assume that that event follows directly from the the event in structure (95a) (perhaps by linking them with the *AFTER* predicate). Note that the *E* variable in structure (95c) denotes that the predicate holds at all times in the event, which is a way of representing states in VerbNet.

#### 8.1.4 Notable Differences and Similarities

Both Jackendoff and Parsons operate with two distinct events here, the first being the causative event where John is the Theme, and the second being the inchoative event, where Bill is the Theme. This is represented by the use of the *CAUSE* and *INCH* functions in the conceptual structures, and *CAUSE* and *BECOME#* in the underlying event formulas. Similarly, VerbNet also makes use of a *CAUSE* predicate, though it lacks any representation of an inchoative event, referring instead to one, single event and its various stages. It also operates with a Patient, rather than a Theme, though these are very much the same kind of role.

While Jackendoff and Parsons refer to the state of being dead, VerbNet instead refers to the state of being alive, by the use of the predicate *ALIVE*. In other words, they use different primitives when it comes to this state of being. The VerbNet structures capture the fact that Caesar is alive before he becomes dead, something which is not represented in Jackendoff's or Parsons' structures. So, in a way, VerbNet captures an inherent condition that says that for something to die it must first be alive.

Both the conceptual structures and the underlying event formulas capture the wanted entailment in this example. The VerbNet structures, on the other hand, does not capture the most important entailment, namely the fact that if someone is killed then that person also dies. This is because

of the different semantic predicates used in the two structures.





## Chapter 9

# Verbs of Putting: *Spray/Load* Verbs

Verbs that fall into this category have been chosen (note that Levin lists *spray/load*-verbs as a subclass of Verbs of Putting) because they can operate in an interesting kind of alternation called the "spray/load" alternation [17, p. 50]. They involve the organization or placement of something in a location, either all over it or in one particular place. Other verbs of this kind include *splatter*, *crowd* and *pile*, for instance. Note that the following section will only be looking at the verb *load*, since the focus will be on the entailment involved with the particular alternation mentioned [17, p. 117].

### 9.1 Load

The verb *load* can be used in at least two different ways which alter the overall meaning. Sentences using the verb typically describe events where something is loaded into or onto some location and the differing meanings depend on the order of these participants in the syntax. If the thing being loaded, let's call it the Theme, is in the direct object position, and the location is included at the end of the sentence in a prepositional phrase (starting with *onto* or *into*, for instance), then it is not really certain that the location was completely filled with the Theme. In other words, there is no sense of "completeness", that the Theme was evenly distributed across the location [15, p. 106].

If, however, the *load* sentence uses the location as the direct object, and the Theme as an argument to a following prepositional phrase (typically a *with*-phrase), then the idea of completeness is fully conveyed. It is now certain that the location is completely filled, or distributed, with the Theme. Consider the sentences in (96) as illustrative examples, where sentence (96a) shows a non-distributive version of *load*, and sentence (96b) shows a distributed one.

- (96) a. *Kim loaded bricks onto the truck.*  
b. *Kim loaded the truck with bricks.*

Here, it is certain that Kim loaded the truck completely with bricks in

sentence (96b), but not in sentence (96a), as the truck might only end up half-full in that sentence. However, it seems logical to imagine that the sense of *load* in (96a) involves the same kind of action as in sentence (96b), except that we don't know if the goal of reaching a completely filled truck is ever reached. But the action done in sentence (96b) would still involve the same kind of action as in sentence (96a); if Kim managed to completely fill the truck with bricks, then she also loaded bricks onto the truck in order to do so [15, p. 172]. As such, it seems logical to assume that the sentence in (96b) entails the one in (96a), and not vice versa.

It is worth noting, however, that WordNet lists these two meanings as two distinct senses for the verb *load* [18]. One involves "filling or placing a load on", which would be the distributive meaning, and the other involves "putting something on a structure or conveyance", which would instead be the non-distributive meaning. These synsets do not share any relations with each other in that database, leading to the idea that there might not be an entailment here after all. However, before concluding anything, the following sections will look at the structures proposed by the three representations under discussion.

### 9.1.1 Jackendoff's Conceptual Structure Representation

In Jackendoff [15, p. 171-174], this issue regarding *load* and similar verbs is discussed. The author mentions that there are two possible ways that the two kinds of events might be connected. Both involves representing the distributive meaning as a causative-inchoative and the non-distributive as involving locative motion along a path. The first method assumes that the the core meaning is the non-distributive one and the other meaning is then simply a more elaborate modification of that core meaning. In other words, the sentence in (96b) can be paraphrased as "Kim filled the truck with bricks by loading bricks onto the truck", where the core meaning lies in the sentence following *by*. Notice that the word *filled* is used here instead, conveying the fact that the truck was completely loaded.

The other method involves the complete opposite. The core meaning is instead said to be the distributive sentence and the other kind is simply an elaboration. Therefore, the sentence in (96a) can be paraphrased as "Kim loaded bricks onto the truck in order to fill the truck with bricks." In other words, Kim loaded the bricks onto the truck so that the goal of filling it is completely reached. Jackendoff does not reach a conclusion as to which of these methods should be applied to his theory. However, we can assume that the first method is the most proper solution here, as Jackendoff has conveniently introduced the function *BY*, which seems to cover the nature of the first example sentence given above. In other words, a conceptual structure for the sentence in (96b) would involve the use of the *BY* function, which would take as its argument the same event structure as for the sentence in (96a).

Consider the proposed structures shown in (97) below, which makes use of the lexical entry that Jackendoff proposes in Jackendoff [15, p. 173]. Notice the use of the superscript *+D* which, as mentioned in section 3.4,

conveys the distributive meaning of the place function. Note also the use of the *PL* function, which is used to represent the word *bricks*, a plural (this was discussed in section 3.5).

- (97) a. [event CAUSE ([thing KIM],  
           [event GO<sub>LOC</sub> ([thing PL<sup>-B,+I</sup> ([thing<sup>+B,-I</sup> BRICK)]),  
           [path TO ([place ON ([thing TRUCK]))]])])]  
   b. [event CAUSE ([thing KIM],  
           [event INCH ([state BE<sub>LOC</sub>  
                       ([thing PL<sup>-B,+I</sup> ([thing<sup>+B,-I</sup> BRICK)]),  
                       [place ON<sup>+D</sup> ([thing TRUCK]))]])])]  
       [BY [event CAUSE ([thing KIM],  
           [event GO<sub>LOC</sub> ([thing PL<sup>-B,+I</sup> ([thing<sup>+B,-I</sup> BRICK)]),  
           [path TO ([place ON ([thing TRUCK]))]])])])]

The structure in (97a) says that Kim causes the bricks to go on a locative path onto (or to on) the truck. This same event structure is used inside the *BY* function in structure (97b). In addition, structure (97b) says that Kim caused the inchoative event where the bricks end up being evenly distributed onto the truck. In other words, Kim loaded the truck completely with bricks by doing the same event in structure (97a), namely by causing the bricks to go onto the truck. Because the structure in (97a) is embedded inside structure (97b), then the correct entailment is captured.

### 9.1.2 Parsons' Underlying Event Representation

Interestingly, Parsons includes the verb *load* both in his list of transitive verbs and in his list of causative-inchoatives [25, p. 266-268]. In the latter list there are two entries for the verb, which both have the same resulting adjective state, namely the state of being *loaded*. So, the only difference between the two entries comes from the example sentences he provides, which are *load the wagon* and *load the hay*. These appear to convey the same kind of meaning as the two sentences provided in (96), where the wagon can be seen as the location and the hay can be seen as the Theme. As such, one entry seems to convey the distributive meaning, while the other the non-distributive meaning. After all, there is presumably no way of loading hay with anything. The question is then what the difference between the two resulting formulas will be.

In the list of transitive verbs, Parsons also lists two entries for *load*, where one is an event and the other is a process. Remember from section 4.4 that a process is still a kind of event, except that it does not culminate, but rather holds. The sentence in (96a) can be seen as a process that could have been stopped at any time, even before a goal of perhaps completely filling the truck was achieved. Such an alternative event would instead be an accomplishment, where the event would culminate the moment the truck was finally fully loaded. The example that Parsons provides, which is *load hay in the barn* supports this, as that sentence could easily be replaced by the one in (96a) and still retain the same event type. That sentence can

be given an alternative progressive form, such as "Kim was loading bricks onto the truck."

Because of these two listings, this section will be looking at the two formulas separately, before reaching a conclusion as to which of them would be the most fitting to use in order to capture the wanted entailment. Utilizing the template that Parsons provides for a transitive sentence [25, p. 266], the resulting formulas for the sentences in (96) are shown in (98). The alternative causative-inchoative formulas are shown further below.

- (98) a.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \text{Hold}(e, t) \ \& \text{Loading}(e) \ \& \text{Agent}(e, \text{Kim}) \ \& \text{Theme}(e, \text{Bricks}) \ \& \text{Onto}(e, \text{Truck})]$   
 b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \text{Cul}(e, t) \ \& \text{Loading}(e) \ \& \text{Agent}(e, \text{Kim}) \ \& \text{Theme}(e, \text{Truck}) \ \& \text{With}(e, \text{Bricks})]$

Assuming that the direct object of each sentence maps into the Theme of the semantic formula, and the following prepositional phrases are each given their own predicate, the formulas in (98) have many differences that result in the fact that there is no entailment between them. Although both are *Loading* events that took place in an interval in the past, formula (98a) is a process that held at a point in time within that interval, while formula (98b) culminated at that time. Also, the Themes are different between the two, being the bricks in (98a), but the truck in formula (98b). Finally, the use of the two predicates *Onto* and *With*, which represent the adjunct prepositional phrases, are different between the two formulas. Because of this, the notion that formula (98b) should entail formula (98a) is not captured.

As such, let's look at the alternative causative structures instead and see what the differences are. These structures are shown in (99) below. Note that the *With* predicate used in formula (99a) (and in formula (98b) as well) is not the same as the instrumental *With* (which was renamed *Instrument*), but rather an ornamental *With*, which says that something is added to the Theme.

- (99) a.  $(\exists e_1)(\exists t_1)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \text{Cul}(e, t) \ \& \text{Agent}(e, \text{Kim}) \ \& (\exists e_2)(\exists t_2) [t_2 \in I \ \& \text{BECOME\#}(\text{Loaded})(e_2) \ \& \text{Theme}(e, \text{Bricks}) \ \& \text{Onto}(e_2, \text{Truck}) \ \& \text{Cul}(e_2, t_2) \ \& \text{CAUSE}(e, e_2)]]$   
 b.  $(\exists e_1)(\exists t_1)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \text{Cul}(e, t) \ \& \text{Agent}(e, \text{Kim}) \ \& (\exists e_2)(\exists t_2) [t_2 \in I \ \& \text{BECOME\#}(\text{Loaded})(e_2) \ \& \text{Theme}(e, \text{Truck}) \ \& \text{With}(e, \text{Bricks}) \ \& \text{Cul}(e_2, t_2) \ \& \text{CAUSE}(e, e_2)]]$

Both formulas involve two events,  $e$  and  $e_2$ , where the former is the causing event, and the latter is the inchoative event. In other words, in both cases, the Agent (*Kim*) does something that creates a state of the truck being loaded ( $e$  causes  $e_2$ ). However, formula (99b) has an additional predicate (*With*) which represents the fact that what the truck is being loaded with is bricks. The *BECOME#* function thereby introduces a state of *being-loaded-with-bricks* which has the truck as its Theme. Formula (99a), lacks the *With* predicate, but it instead uses the predicate *Onto*, with *Truck* as its second

argument. Also, a different Theme for the  $e_2$  event is used in formula (99a) (the bricks, rather than the truck).

These small inequalities result in some rather subtle differences in meaning. While formula (99b) focuses on the truck as the Theme, and how it ends up having the state of *Being-loaded-with-bricks*, formula (99a), on the other hand, focuses on the bricks being the Theme, and says that they end up in a state of *Being-loaded-onto-the-truck*. In other words, the formulas seem to represent two distinct senses of the adjective *loaded*, formula (99a) saying that the bricks have been loaded onto the truck, and formula (99b) saying that the truck has been loaded with bricks. While it is correct to say that the truck is completely loaded in formula (99b), it is likewise correct to say that the bricks are also completely loaded in formula (99a).

The question is then how the loaded truck is related to the loaded bricks. How can the fact that the bricks that are put onto the loaded truck in formula (99b) are also in the state of having been loaded onto the truck? In the current formulas there is no representation of the fact that formula (99b) also involves the bricks being loaded onto the truck, which is represented in formula (99a). Therefore, formula (99b) does not entail formula (99a), which, as suggested, should be the case. This observation might, however, suggest that this entailment is incorrect after all, as there is clearly a difference between loading bricks onto the truck and the truck itself becoming loaded with bricks.

Still, following the reasoning of Jackendoff's structures (shown in structure (97)) it seems logical to assume that for a truck to become loaded with bricks there must be an event where bricks are actually loaded onto the truck. But Parsons' current formulas do not include such a relationship between the events. If they did, then the formula in (99b) would include some representation of the fact that the bricks also become loaded (the structure in (99a)). So, unlike Jackendoff, Parsons proposes that there is no entailment present between the two sentences here (this is clear from both the structures in (98) and (99)).

### 9.1.3 VerbNet's Semantic Representation

*Load* is a member of only one class in VerbNet, namely *spray-9.7-2*. From its superclass, the thematic roles Agent, Theme and Destination are available, and the Destination must be both a location and not a region. The subclass that *load* is a member of also adds another restriction to the Theme role, saying that it must be concrete.

Given the frames available, where sentence (96a) has the phrase structure *NP V NP PP.Destination* and (96b) has the phrase structure *NP V NP.Destination PP.Theme*, the resulting semantic representations are shown in structure (100), where *A* is the Agent, *T* is the Theme, and *D* is the Destination. Note that in the original structure for (100a) a predicate called *PREP* is used as a placeholder for whatever preposition is used in the sentence [30, p. 389]. In this case the preposition is *onto*, so the predicate ends up being *ONTO*. *PREP* is, in other words, a sentence specific predicate, or variable predicate, and it is added as a subscript when it

appears.

- (100) a. MOTION(DURING(E), Bricks<sub>T</sub>) &  
 NOT(ONTO<sub>PREP</sub>(START(E), Bricks<sub>T</sub>, Truck<sub>D</sub>)) &  
 ONTO<sub>PREP</sub>(END(E), Bricks<sub>T</sub>, Truck<sub>D</sub>) & CAUSE(Kim<sub>A</sub>, E)  
 b. MOTION(DURING(E), Bricks<sub>T</sub>) &  
 NOT(LOCATION(START(E), Bricks<sub>T</sub>, Truck<sub>D</sub>)) &  
 LOCATION(END(E), Bricks<sub>T</sub>, Truck<sub>D</sub>) & CAUSE(Kim<sub>A</sub>, E)

The only difference between the two structures is the use of either the predicate *ONTO* or *LOCATION*. The order and thematic role of the participants remain the same in both; the bricks are the Theme and the truck is the Destination. Structure (100a) says that the bricks are in motion during the event because of Kim, and that the bricks are not on the truck at the start of the event, but are at the end. Structure (100b) pretty much says the same thing, except for saying that the bricks are not at the location of the truck at the start, but are at the end of the event.

So, given the fact that two different predicates are used here, there is no entailment present between the two formulas. If the predicate had been the same, however, the structures would have entailed one another (they would be the same), so structure (100a) would incorrectly entail structure (100b). While the structure in (100a) seems like a logical way of representing the given sentence, the structure in (100b) seems a little less natural, because it doesn't say anything about the state of the truck. The fact that the truck should end up in a state of being completely loaded with bricks is not conveyed in the current structure.

A solution might have been to use the predicate *STATE* and say that the state of the truck at the end or as a result of the event is of being loaded. The additional predicate might look like this: *STATE(RESULT(E), Loaded<sub>ENDSTATE</sub>, Truck<sub>D</sub>)*, which uses the verb specific argument *EndState* (see section 5.3.3). Using this additional predicate in conjunction with the use of the same predicate (such as *LOCATION*), the wanted entailment would have been captured.

#### 9.1.4 Notable Differences and Similarities

All three frameworks appoint Kim as the causing Agent; as the first argument to the *CAUSE* function in the conceptual structures, as the Agent of the causing event in the underlying event formulas, and as an argument to the *CAUSE* predicate in the VerbNet structures. Also, in almost all the structures the Theme is said to be the bricks, while the location or the place where they end up is the truck. Even though different functions are used in the core structures in the conceptual structure representation (*BE<sub>LOC</sub>* and *GO<sub>LOC</sub>*), the bricks are always used as the first argument, making it the Theme. In VerbNet, as well, the Theme is always the bricks, even though *the truck* acts as the direct object in sentence (96b).

However, in Parsons formula for the second sentence, shown in (99b), the role of Theme is appointed to the truck instead, while the bricks appear

as an argument to the *With* predicate. The reason for this stems from the mapping from the syntactic structure of the sentence to the resulting semantic formula. The direct object is said to be the Theme, while any participants appearing in additional prepositional phrases are added to corresponding prepositional predicates. Because of this, there is a shift in focus between formulas (99a) and (99b), where in the former the focus of the second event is the bricks, while it is the truck in the latter.

The Jackendoff structure in (97a) focuses on the bricks arriving at a location, by moving along a path. In contrast, the formula that Parsons provides for the same sentence, seen in formula (99a), instead centers on the fact that the bricks ends up having the state of being *loaded*. If the conceptual structure had approached the matter in the same way, then the *GO* function would instead be under the identificational semantic field (using the *IDENT* subscript), where the bricks would end up having the property of being loaded (similar to the way *kill* and *die* was handled in section 8.1). Similar to the Jackendoff approach, the VerbNet structure for the same sentence (seen in structure (100a)) also focuses on the fact that the bricks end up on the truck, rather than saying that the state of the bricks is that of being loaded.

Because the focus in (99b) is on the state of the truck becoming loaded, there is a difference between that structure and those that Jackendoff and VerbNet propose. In neither of the latter two representations is there a mention of the state of the truck, or of it having any sort of property. The focus is still on the bricks arriving at the location of the truck.

The structures in (97) show how the *PL* function is used to effectively represent plurality, a concept that does not have much focus in the other two structures. While the only difference between a *brick* and *bricks* in Parsons' formulas and the VerbNet structures lies in the name of the participant in question, there is a lot more going on in the conceptual structures. However, when it comes to the wanted entailment for the current example there really isn't any important difference between the structure for *bricks* used and just using a thing function called *BRICKS* (without the *PL* function), as long as the names are consistent between the two sentence structures.

So, while the Parsons' formulas center on the fact that the state of either the truck or the bricks ends up as *loaded* (with the additional information, such as *with-bricks* and *on-the-truck*), the other two structures instead say that the location or place that the bricks occupy is the truck (there is no mention of the participants' identificational states). However, because of differences between the structures for the two sentences, only the conceptual structures capture the entailment introduced at the start of this section, even though the VerbNet structures also come rather close to capturing it. It seems the underlying event formulas maintain that there is a clear difference between the two senses of being loaded, while in the conceptual structures, and in the VerbNet structures (to a lesser degree), a relation between them is indeed proposed.





## Chapter 10

# Verbs of Ingesting

These verbs involve the consumption of something by some entity. This chapter be looking at the verb *eat*, though other similar verbs include *devour*, *chew* and *drink*. *Eat* is interesting because it can operate in both transitive and intransitive sentences (something that *devour* can't, for instance), so the focus will be on the entailment relation between such sentences involving *eat* [17, p. 213].

### 10.1 Eat

Although WordNet proposes that the two different uses of the verb *eat* indicate two different senses, namely the notion of "taking in a meal" for the intransitive and "taking in solid food" for the transitive [18], let's assume in this example that these do in fact represent the same sense (VerbNet has only one class for *eat*, for instance). The only difference is the fact that the thing being eaten is known in the transitive use, but unknown in the intransitive.

Essentially, we would want a semantic representation to capture the fact that a transitive sentence like the one in (101a) entails the similar intransitive sentence in (101b) below, but not vice versa.

- (101) a. *Mary ate a cake.*  
b. *Mary ate.*

In other words, if Mary ate a cake, then it is also a fact that she simply ate, but if she only ate, then it may not be true that what she ate was a cake.

#### 10.1.1 Jackendoff's Conceptual Structure Representation

In Jackendoff's framework these two sentences would have the conceptual structures shown in (102) below [15, p. 253]. Notice the use of a semantic function called *MOUTH-OF*, which was not mentioned in the chapter on Jackendoff's theory (chapter 3). The function is a sort of body-part identifier which essentially returns the mouth of some thing. It is similar to the *PART* function, which also identifies that a thing is a part of another thing. So, alternatively, we could have used the structure [*thing* *MOUTH*

*PART ([thing MARY])*] here instead, although the structure used here is what Jackendoff originally proposes.

- (102) a. [event CAUSE ([thing MARY],  
[event GO<sub>LOC</sub> ([thing CAKE],  
[path TO ([place IN  
([thing MOUTH-OF ([thing MARY]))]))])] ]  
b. [event CAUSE ([thing MARY],  
[event GO<sub>LOC</sub> ([thing ],  
[path TO ([place IN  
([thing MOUTH-OF ([thing MARY]))]))])] ]

These structures are both causative events where *Mary* is an Agent causing a locative *GO* event involving a *cake* in structure (102a) and some unknown thing in structure (102b). This thing that was eaten goes on a path into a place (indicated by the *TO* and *IN* functions), where the place in question is the mouth of *Mary*. So, the only difference between these two structures is the fact that the argument to the *GO<sub>LOC</sub>* function is known to be a cake in structure (102a), but is unknown (unspecified) in structure (102b).

Because of this, it is easy to see that the correct entailment is captured here, since structure (102b) is the same as structure (102a), except with a missing specification of the thing argument in the *GO<sub>LOC</sub>* function. This becomes perhaps more apparent if we take a look at the alternative flat structures, shown in (103). Note that *e* is the causing event, *e<sub>2</sub>* is the *GO<sub>LOC</sub>* event, *t* is *Mary*, *t<sub>2</sub>* is the thing being eaten, *t<sub>3</sub>* is the mouth of *Mary*, *p* is the path the thing being eaten travels, and *pl* is the place inside of *Mary*'s mouth.

- (103) a. [CAUSE(*e*) & ARG1(*e*, *t*) & ARG2(*e*, *e<sub>2</sub>*) & GO<sub>LOC</sub>(*e<sub>2</sub>*) & ARG1(*e<sub>2</sub>*, *t<sub>2</sub>*) & ARG2(*e<sub>2</sub>*, *p*) & TO(*p*) & ARG1(*p*, *pl*) & IN(*pl*) & ARG1(*pl*, *t<sub>3</sub>*) & MOUTH-OF(*t<sub>3</sub>*) & ARG1(*t<sub>3</sub>*, *t*) & MARY(*t*) & CAKE(*t<sub>2</sub>*)]  
b. [CAUSE(*e*) & ARG1(*e*, *t*) & ARG2(*e*, *e<sub>2</sub>*) & GO<sub>LOC</sub>(*e<sub>2</sub>*) & ARG1(*e<sub>2</sub>*, *t<sub>2</sub>*) & ARG2(*e<sub>2</sub>*, *p*) & TO(*p*) & ARG1(*p*, *pl*) & IN(*pl*) & ARG1(*pl*, *t<sub>3</sub>*) & MOUTH-OF(*t<sub>3</sub>*) & ARG1(*t<sub>3</sub>*, *t*) & MARY(*t*)]

Again, there is only one difference between the two structures, namely the presence of the function *CAKE(t<sub>2</sub>)*, which indicates that the thing being eaten is a cake. This is missing in structure (103b), but present in structure (103a). Because structure (103b) is made up of the exact same conjuncted functions as structure (103a), except that it is missing one, structure (103a) entails structure (103b), but not vice versa.

### 10.1.2 Parsons' Underlying Event Representation

In the theory of underlying events, the two sentences have the formulas shown in (104) [25, p. 260,265].

- (104) a.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Eating}(e) \ \& \ \text{Agent}(e, \text{Mary}) \ \& \ \text{Theme}(e, \text{Cake})]$   
 b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Eating}(e) \ \& \ \text{Agent}(e, \text{Mary})]$

Here, the sentences are both understood as *Eating* events, which culminated at a point in time that happened in some time interval in the past. Also in both, the Agent of the event is *Mary*. However, formula (104a) is different from formula (104b) because it has an additional predicate that specifies what the Theme of the event is, namely a *Cake*. This predicate is of course missing from (104b). Because the formula in (104b) is the exact same as the one in (104a), but without the Theme predicate, formula (104a) entails formula (104b), and formula (104b) does not entail formula (104a). As such, the correct entailment is captured in Parsons' framework.

### 10.1.3 VerbNet's Semantic Representation

As mentioned, the verb *eat* is only a member of a single class in VerbNet, namely *eat-39.1-1*. This class allows both the transitive and the intransitive use of the verb, represented by two frames that have the phrase patterns *NP V NP* and *NP V*, respectively. These two frames also make use of the same semantic predicate when representing the meaning. The structures are shown in (105) below, where the *A* subscript constitutes the Agent, and *P* the Patient.

- (105) a. *TAKE\_IN*(*DURING*(*E*), *Mary<sub>A</sub>*, *Cake<sub>P</sub>*)  
 b. *TAKE\_IN*(*DURING*(*E*), *Mary<sub>A</sub>*, ?*Patient*)

We can assume the *TAKE\_IN* predicate represents the fact that one participant (the Patient in this case) is consumed by (or taken in by) some other participant (the Agent here). There is only one difference between the two structures, namely the specification of the Patient role. In structure (105a) it is known to be a cake, but in structure (105b) it is unknown. Following the approaches VerbNet has concerning verbal inference, the correct entailment is perfectly captured. In structure (104b) the Theme is simply removed, so an *Eating* event, in that case, seems to involve only *Mary*, not any unknown Theme.

### 10.1.4 Notable Differences and Similarities

In all three representations, *Mary* is the Agent, while the cake is the Theme or Patient. Also in all three, the only difference between the two structures is the fact that the thing being eaten is a cake in the transitive sentence, and unknown in the intransitive sentence. Consequently, the wanted entailments are captured by all three representations.

However, while the conceptual structures and the VerbNet structures both indicate that there is certainly a thing present that is being eaten (constituted by the empty *thing* function in structure (102b) and the unknown Patient in structure (105b)), there is no mention of this thing

in the underlying event formula in (104b). This illustrates how the total removal of predicates in Parsons' formulas result in there not being any implied unknown participant (in this case, the thing being eaten).

This is related to the issue of empty participants, where a possible *eat*-sentence might be *Mary ate nothing*, for instance. Is this still an event of eating, except without any thing being eaten, or is it not an event at all? Because no Theme is mentioned in formula (104b), it seems that the former case is possible in Parsons' theory, and not in the other two, because they require a thing being eaten. In other words, an event of eating requires a thing being eaten in Jackendoff's theory and in VerbNet, but not in Parsons' framework.

Structure (102) mentions the fact that the cake enters through Mary's mouth, a notion that is missing in the other two representations. VerbNet, on the other hand, uses the more general predicate *TAKE\_IN*, which does seem to constitute the same kind of meaning, though the Theme need not specifically enter through the mouth. However, that is where eating takes place, so the *MOUTH-OF* function in Jackendoff's structures is still appropriate. The underlying event formula does not say anything about this, however.

## Chapter 11

# Verbs of Motion

Many verbs involve the motion of some entity, which is either moved by some force or other entity, or by itself. Section 11.1 below will be looking at the verb *jump* for the former case, a verb that can be used in either a transitive or an intransitive sentence. For the latter case (section 11.2), the example verb will be *climb*, which can also be used either transitively or intransitively. A notion of interest here is whether or not we know where the thing moving starts or ends up, or if it just stays in relatively the same place. Other verbs of motion include, for instance, *run*, *escape* and *dance* [17, p. 263].

### 11.1 Jump

The verb *jump* is allowed in both a transitive and intransitive construction, as mentioned. In an intransitive use it is the subject that undergoes the action, i. e. the one who does the jumping. But in an active transitive use it is instead typically the direct object who does this action. It can also mean that the subject jumps over the direct object, but the focus in this section will be on the former case. When it is the direct object that performs the jumping the subject acts as an "instigator" [15, p. 151], the participant who initiates the jumping. The question is then whether or not this instigator, or causer, also performs the jumping action along with the direct object. Consider the sentences in (106), which involves the two participants *Tom* and *the horse*.

- (106) a. *Tom jumped the horse.*  
b. *Tom jumped.*  
c. *The horse jumped.*

The sentence in (106a) seems to have two different interpretations. Either Tom caused the horse to jump by simply standing still (not following the action and doing the same jump movement) or he is actually riding the horse, in which case he would also be jumping along with it. In other words, (106a) either only entails the sentence in (106c), or it entails both sentences (106b) and (106c).

Adding credibility to this, WordNet proposes, among a list of fifteen, two relevant verb senses for *jump*, one described as "moving forward by leaps and bounds" and another as "cause to jump or leap" [18]. The latter of these senses has a cause relation to the former, indicating that a transitive sentence using *jump*, involving a direct object, would entail the alternative intransitive construction where the direct object instead acts as the subject. However, there is still the question of whether or not the causer (Tom in this case) moves along in the action or not.

Example sentences used for the *cause to jump or leap* synset in WordNet include *The trainer jumped the tiger through the hoop*, where the trainer would typically not move along with the tiger, but the sentence *The men jump the horses across the field* is also included, which seems to most logically involve both the men and the horses jumping across the field. It seems, therefore, that this sense should be further divided into two more distinct senses; one where the causer and the direct object moves in unison, and the other where only the direct object moves.

Assuming sentence (106a) has a meaning resulting in the movement of both Tom and the horse, it entails both sentences (106b) and (106c), but, on the other hand, assuming a meaning where only the horse moves, sentence (106a) entails only sentence (106c). A semantic framework representing the verb *jump* should take this distinction into account, namely the fact that it can have two different senses, and that this results in different entailments.

### 11.1.1 Jackendoff's Conceptual Structure Representation

When *jump* is used as a causative (as it is in sentence (106a)), Jackendoff proposes the use of the *CAUSE* function, where Tom would be the causing Agent and the effect would be the jumping of either both participants, or the horse only [15, p. 151]. In Jackendoff [15], however, there is no clear lexical entry for the verb *jump*, so there is some uncertainty here as to what the final conceptual structures should look like. However, assuming that *jump* has a similar meaning to other verbs of motion that does not involve a specific path, like *dance*, *wiggle* and *wave*, for instance, the correct function to use here seems to be *MOVE* [15, p. 88-89]. As explained in section 3.2.1, the function simply takes a single argument, which is the thing doing some movement.

Note, however, that if we added a prepositional phrase to the sentences in (106), such as *The horse jumped across the field*, the meaning would instead involve a locative *GO* function specifying a path, represented by the function *ACROSS*. This would result in the structure shown in (107) for the sentence mentioned.

(107) [event  $GO_{LOC}$  ([thing HORSE], [path *ACROSS* ([thing FIELD])])]

We could, then, perhaps assume that a pure use of the verb *jump* (such as the sentence in (106b)) have this same structure, but with an unspecified path. This would entail that the thing undergoing the action ends up at a different place than what it originally was, however, a notion that might not always be the case when someone is jumping (they may end up in the exact

same place at the end of it). Even so, both these alternatives (using either *MOVE* or *GO<sub>LOC</sub>* functions) will be considered for this example, in order to see if one is better than the other in capturing the wanted entailment.

At the same time, the idea that both participants might be performing the jump action would intuitively involve the conjunction of two jump events as the effect of the *CAUSE* function. This would be done in a similar fashion to the conjunction of the *TO* and *FROM* functions for specifying a path, seen in the example in (36b), in section 3.3.

Also, if the function *MOVE* is used, there is the question of how such a structure would differ from other structures which involve the use of other kinds of verbs. How, for instance, would the difference between someone dancing and someone jumping be captured if only the function *MOVE* is used to represent both of them? The solution is probably to use an additional manner function in conjunction with *MOVE*, as Jackendoff has done with the lexical entry for the verb *wiggle*, for instance [15, p. 274]. In that example, though, it was used primarily to represent an adverb, such as in *wiggle fast*. However, assuming this is a way to make the movement verbs distinct, a manner function such as *JUMPINGLY* might be used in this case.

Structures (108) and (109) show two kinds of conceptual representations that one can assume for the given sentences, based on Jackendoff's provided framework. Structure (108) uses the *GO<sub>LOC</sub>* function, while structure (109) uses the *MOVE* function in conjunction with a *JUMPINGLY* manner function. Note that for both sets of structures, there are two different structures for the same sentence in (106a), representing the two distinct senses mentioned. These are shown as *a.1* and *a.2*, respectively.

- (108) a.1 [event CAUSE ([thing TOM],  
                  [event GO<sub>LOC</sub> ([thing HORSE], [path ]))])  
      a.2 [event CAUSE ([thing TOM],  
                  [event GO<sub>LOC</sub> ([thing HORSE], [path ])]  
                  GO<sub>LOC</sub> ([thing TOM], [path ])) )]  
      b. [event GO<sub>LOC</sub> ([thing TOM], [path ])]  
      c. [event GO<sub>LOC</sub> ([thing HORSE], [path ])]
- (109) a.1 [event CAUSE ([thing TOM],  
                  [event MOVE ([thing HORSE])  
                  [manner JUMPINGLY]])]  
      a.2 [event CAUSE ([thing TOM],  
                  [event MOVE ([thing TOM])  
                  MOVE ([thing HORSE])  
                  [manner JUMPINGLY]])]  
      b. [event MOVE ([thing TOM])  
                  [manner JUMPINGLY]]]  
      c. [event MOVE ([thing HORSE])  
                  [manner JUMPINGLY]]]

There doesn't seem to be any difference between the two variations with regards to entailment. Structure (108a.<sub>1</sub>), which represents the meaning where only the horse goes on a locative path, correctly entails only structure (108c.) and not structure (108b.), because structure (108c.) is the same as the effect event in structure (108a.<sub>1</sub>). Similarly, structure (109a.<sub>1</sub>), which represents the idea that only the horse moves, entails structure (109c.) for the same reason; the latter structure is used as an argument to the former. In addition, structure (108a.<sub>2</sub>) entails both structures (108b.) and (108c.) because both of those structures are part of structure (108a.<sub>1</sub>), as a conjuncted event using two *GO<sub>LOC</sub>* functions. Likewise, structure (109a.<sub>2</sub>) entails structures (109b.) and (109c.) because the latter two also take part in the more complex structure (109a.<sub>1</sub>). So, assuming that one of the sets of structures is the correct to use for the sentences in question, the entailments are indeed captured in Jackendoff's theory.

### 11.1.2 Parsons' Underlying Event Representation

Although Parsons does not include *jump* in his list of verbs in Parsons [25, p. 264-268], let's assume the intransitive variant has a similar meaning to the verb *move*, which is included in his list of intransitive verbs. For that verb, the subject in question is proposed to be either a Theme or an Agent-Theme. We can also further assume that the transitive variant of *jump* represents a causative event which brings about the existence of the jumping event represented by the intransitive use. As such, the underlying event formulas for the sentences in (106) are shown in (110) below. As in the conceptual structures, *a*<sub>1</sub> and *a*<sub>2</sub> both represent the sentence in (106a), while *b*. and *c*. represent the other two sentences

- (110) a.<sub>1</sub>  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Agent}(e, \text{Tom}) \ \& \ (\exists e_2)(\exists t_2) [t_2 \in I \ \& \ \text{Cul}(e_2, t_2) \ \& \ \text{Jumping}(e_2) \ \& \ \text{Theme}(e_2, \text{Horse}) \ \& \ \text{CAUSE}(e, e_2)]]$
- a.<sub>2</sub>  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Jumping}(e) \ \& \ \text{Agent-Theme}(e, \text{Tom}) \ \& \ (\exists e_2)(\exists t_2) [t_2 \in I \ \& \ \text{Cul}(e_2, t_2) \ \& \ \text{Jumping}(e_2) \ \& \ \text{Theme}(e_2, \text{Horse}) \ \& \ \text{CAUSE}(e, e_2)]]$
- b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Jumping}(e) \ \& \ \text{Theme}(e, \text{Tom})]$
- c.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Jumping}(e) \ \& \ \text{Theme}(e, \text{Horse})]$

There are a couple of differences between formulas (110a.<sub>1</sub>) and (110a.<sub>2</sub>). While both include two events, *e* and *e*<sub>2</sub>, where *e*<sub>2</sub> is a *Jumping* event with *Horse* as its Theme, and both say that *e* caused *e*<sub>2</sub>, there is a difference in what kind of event *e* is and what role *Tom* plays. Although it seems logical to say that *Tom* is the causing Agent in formula (110a.<sub>1</sub>) that brings about the existence of *e*<sub>2</sub>, the fact that both of the participants must undergo a jumping movement seems a little harder to convey in formula (110a.<sub>2</sub>). In formula (110a.<sub>1</sub>), *Tom* does not jump, so he is simply the Agent, and the



nature of  $e$  is left unspecified. Formula (110a.2), on the other hand, must take into account that Tom also jumps.

We can assume that the correct way is to say that  $e$  is a jumping event as well, similar to  $e_2$ , where Tom is the Theme as well as the Agent. In other words, Tom causes himself to jump, as well as causing another event where the horse also jumps. So, formula (110a.2) has an additional predicate saying that  $e$  is a *Jumping* event, in addition to a predicate that says that Tom is an Agent-Theme (he has the features of both an Agent and a Theme). The formulas in (110b.) and (110c.), on the other hand, are not as complex, consisting instead of only one event, and differing only in their specification of what the Theme is.

Given these representations, it is clear that formula (110a.1) correctly entails formula (110c.), because formula (110c.) is exactly the same as the inner formula in (110a.1). In other words, event  $e$  in formula (110c.) is the same as event  $e_2$  in formula (110a.1). The fact that formula (110a.2) entails both formulas (110b.) and (110c.) is a little harder to see, however. Although formula (110c.) is entailed because of the exact same reason why formula (110a.1) entails it, formula (110b.) uses a different predicate for Tom than what is used in formula (110a.2). Because of this, it seems at first that the wanted entailment is not captured here. However, if we assume that *Agent-Theme*( $e$ , Tom) means basically the same as *Agent*( $e$ , Tom) & *Theme*( $e$ , Tom), the correct entailment is indeed captured. The  $e$  event in formula (110b.) is exactly the same as the  $e$  in formula (110a.2), except that it does not include a predicate saying that Tom is also the Agent. So, assuming these are the correct formulas, Parsons' theory also captures the wanted entailments in this example.

### 11.1.3 VerbNet's Semantic Representation

VerbNet lists two classes that have *jump* as a member, namely *calibratable\_cos-45.6-1* and *run-51.3.2-2-1*. However, let's assume that the latter class is the correct for the meaning in question, since the other class represents a sense where some attribute moves along a scale (it changes its value rather than physically moving). *Run-51.3.2-2-1* uses an animate Theme that might move in a concrete Location, caused by some animate Agent. *Jump* is also used in most of the example sentences given for the frames, and they all carry the same kind of meaning as the sentences discussed in this chapter. In fact, the sentences in (106) were borrowed from this class in VerbNet.

The frames available in the *run-51.3.2-2-1* class, and its superclasses, result in the semantic structures shown in (111), where  $T$  is the Theme and  $A$  is the Agent. Note, however, that there are two possible frames available for the transitive sentence in (106a), which has the phrase structure *NP V NP*. Both frames come from the superclass *run-51.3.2-2*, and there is no difference between their syntactic structures (both use *Agent V Theme*). Their semantic structures differ quite a bit, though, and these two structures are shown in (111a.1) and (111a.2), respectively. The structures for the other two sentences are shown in (111b.) and (111c.), using a frame available in the superclass *run-51.3.2*.

- (111) a.<sub>1</sub> MOTION(DURING(E0), Horse<sub>T</sub>) & CAUSE(Tom<sub>A</sub>, E0) & EQUALS(E0, E1) & MOTION(DURING(E1), Tom<sub>A</sub>)  
 a.<sub>2</sub> MOTION(DURING(E), Horse<sub>T</sub>) & CAUSE(Tom<sub>A</sub>, E)  
 b. MOTION(DURING(E), Tom<sub>T</sub>)  
 c. MOTION(DURING(E), Horse<sub>T</sub>)

Since there is no difference in syntax between the two frames used in structures (111a.<sub>1</sub>) and (111a.<sub>2</sub>) it is a little hard to see which version should be chosen, based solely on the sentence given. It seems the frame must be chosen depending on what the intended meaning is; whether both participants moves or only the Theme. In structure (111a.<sub>1</sub>) there are two distinct events (*E0* and *E1*), and in each of these one of the participants is in motion during the event. In *E0* it is the horse that is moving, and in *E1* it is Tom. These two events are said to take place at the same time, represented by the *EQUALS* predicate. Also, Tom causes *E0*.

Structure (111a.<sub>2</sub>) is very similar to structure (111a.<sub>1</sub>), but it only involves a single event, which is also caused by Tom, and where the horse moves in its during stage. So, in contrast to structure (111a.<sub>1</sub>), only the horse moves, not Tom, but he is still the causer of the event. Structures (111b.) and (111c.) are the same, except that the Themes are different. In both cases the Theme moves during the event.

The predicate used in structure (111c.) is also present in structure (111a.<sub>2</sub>) (with the same arguments), so the latter entails the former. The predicates in structures (111b.) and (111c.) are also present in structure (111a.<sub>1</sub>), so structure (111a.<sub>1</sub>) also correctly entails structures (111b.) and (111c.). There is still the issue of how the intended meaning is chosen based on the given sentence, but assuming that the two structures in (111a.<sub>1</sub>) and (111a.<sub>2</sub>) represent the two different senses discussed, the correct entailments are captured in VerbNet.

#### 11.1.4 Notable Differences and Similarities

For the sentence in (106a), when the sense involves only the movement of the horse, Tom is said to be the Agent in all the three representations (in the *a.<sub>1</sub>* structures). In structures (108a.<sub>1</sub>) and (109a.<sub>1</sub>), Tom acts as the first argument to the *CAUSE* function, making him the cause or Agent, while in formulas (110a.<sub>1</sub>) and (111a.<sub>1</sub>), Tom is directly appointed the role of Agent. In every structure given, except the ones representing the sentence in (106c), the horse acts as the Theme, as it is the first argument to either the *GO<sub>LOC</sub>* function or the *MOVE* function for the conceptual structures in (108) and (109), and is specified as this role in both Parsons' and VerbNet's structures, seen in (110) and (111), respectively. So, every time the horse appears, it occupies the role of Theme.

Tom, on the other hand, acts as both the Theme and the Agent in the structure representing the sense of the sentence in (106a) involving the movement of both participants. This is at least the case in the structures provided by Jackendoff and Parsons, where Tom is both an argument to the *CAUSE* and the *GO<sub>LOC</sub>* functions in the former, and is specified as

an Agent-Theme in the latter. This dual labeling of roles is not present in the structure given by VerbNet, as Tom is still only the Agent, while the horse is the Theme, in structure (111a.<sub>1</sub>). In addition, Tom also acts as just the Theme in the structures for the sentence in (106b), in all the three representations. In other words, he either acts as the argument to the *GO<sub>LOC</sub>* or *MOVE* functions in the conceptual structures, or is directly specified as the Theme in Parsons' and VerbNet's formulas. So, while Jackendoff and Parsons pretty much agree on the thematic roles given to the participants in these example sentences, the VerbNet structures does not say that Tom is the Theme in structure (111a.<sub>1</sub>), even though he is also said to be moving. This illustrates both the fact that participants can only occupy one role in VerbNet, but also the fact that thematic roles can appear in any predicate. So, even though the Theme is typically the role that undergoes an action or is changed or moved somehow, the fact that this can also happen with an Agent is possible in VerbNet, exemplified by the fact that the Agent appears in the *MOTION* predicate in structure (111a.<sub>1</sub>). In other words, the predicates provide no restrictions on what role the arguments must occupy.

For the conceptual structures representing the sentence in (106a), Jackendoff says that the overall event is a causative, which involves either one or two embedded *GO<sub>LOC</sub>* event or events. Parsons, on the other hand, says that both the possible senses for that sentence involve two separate events, where the first causes the other. In contrast, VerbNet proposes that there is only one event if only the horse moves, but two distinct events if both participants move. These two events are said to take place at the same time, a notion that is not present in the given conceptual structures or underlying event formulas. Though Parsons' formulas say that the two events took place in the same time interval, they culminate at different points in time ( $t$  and  $t_2$ ). Because Jackendoff does not refer to any concept of time in his structures, there is no mention of whether or not the events took place at the same time or not. So, for VerbNet's structure it is clear that what Tom did to cause the horse to jump, and the horse's jumping movement, took place at the exact same time. This may not have been the case for the other two representations. For those structures, what Tom did might have taken place prior to the horse jumping, for instance. However, Parsons' structures could have conveyed the same information as in the VerbNet structures, if  $t$  and  $t_2$  were said to be the same points in time. Then, the two events would have taken place within the same interval of time, and would have culminated at the same time. The point in time when both of them started would be unclear, however.

The fact that one or both of the participants move is captured in all three representations, if we assume that a Theme is said to be moving in a *Jumping* event in Parsons' theory. The idea is conveyed much more directly in Jackendoff and VerbNet's structures, however, where the former either says that the participant goes on a locative path, or simply moves, while the latter says that during the event the participant is in motion. However, the fact that an event of jumping should be different from an event of hopping, for instance, would be captured in Parsons' theory, through the use of

the predicates *Jumping* and *Hopping* to describe the event. The structures that Jackendoff and VerbNet provide are much more general, capturing a meaning shared by many kinds of verbs. Though the conceptual structures do have the manner option available (as used in structures (109), VerbNet does not include any such specifics. In other words, while the underlying event theory makes all events, specified by different verbs, distinct, the other two theories focus on the generality of their structures, in order to capture the fact that many verbs involve the same kind of meaning.

## 11.2 Climb

For this example, the focus will be on a kind of diathesis alternation where one participant either appears as a direct object or as part of a prepositional adjunct. This is also called the preposition drop alternation [17, p. 43]. The subject is the same throughout and the other participant is then the entity that is climbed. Consider the examples in (112), where sentence (112a) is a standard transitive version, sentences (112b) and (112c) use different prepositional phrases, while sentence (112d) is the simple intransitive variant. In all cases the subject is *Jane* and the other participant (the thing being climbed) is the *mountain*.

- (112) a. *Jane climbed the mountain.*  
       b. *Jane climbed up the mountain.*  
       c. *Jane climbed down the mountain.*  
       d. *Jane climbed.*

In sentence (112a) it is clear that Jane completely climbed the mountain, and thereby reached the top of it, but this may not be the case in the other sentences. In sentence (112b) she might only reach halfway up the mountain, while in sentence (112c) she might have started climbing down the mountain from some midway point (not from the top). Sentence (112d) doesn't even mention the mountain, so the activity can have been done anywhere.

However, in sentence (112a) it is also clear that in order for Jane to have climbed the mountain, she must also have done the same activity as in sentence (112b), namely climbing up the mountain. The idea that she also climbed down the mountain need not be the case; she could have climbed straight up to the mountain top. So, effectively, sentence (112a) entails sentence (112b), but it does not entail sentence (112c). In addition, the three sentences in (112a), (112b) and (112c) all involve the activity of climbing, meaning that they all entail the sentence in (112d). As such, a semantic framework would need to account for the different meanings arising from the use of the prepositional phrases in sentences (112), and the differences between the transitive and the intransitive forms.

### 11.2.1 Jackendoff's Conceptual Structure Representation

In Jackendoff [15, p. 76-77] the lexical entry for the verb *climb* is listed. According to Jackendoff, the differences between the sentences in (112) depend on the specifics of the path that the subject goes on, hence all the sentences are represented as locative GO events. The conceptual structures for the sentences in (112) are shown in (113) below, where Jane goes on a locative path. Notice the use of the place function *TOP-OF*, which is used to describe the top of the mountain.

- (113) a. [event GO<sub>LOC</sub> ([thing JANE],  
[path TO ([place TOP-OF ([thing MOUNTAIN]))])] ]  
b. [event GO<sub>LOC</sub> ([thing JANE],  
[path UP ])]  
c. [event GO<sub>LOC</sub> ([thing JANE],  
[path DOWN ])]  
d. [event GO<sub>LOC</sub> ([thing JANE],  
[path ])]

Jackendoff maintains that when *climb* is used with a direct object the fact that the top of this object is reached is true, so the structure in (113a) is the correct to use. However, when *climb* is used with a prepositional phrase instead (as in sentences (112b) and (112c)), the place reached is unknown. As such, the path function is given no argument (which would be the place that was reached, if it was known), though the name of the path function is still specified. For the intransitive sentence, the path climbed is unknown altogether, so the name of the path function is left unspecified. However, it is still true that someone goes on some path when they climb.

The fact that the three structures in (113a), (113b) and (113c) should entail structure (113d) is achieved, because structure (113d) is embedded inside them. This is perhaps more apparent when looking at the flat structures, shown in structures (114). Here, all the conjuncted functions that are present in structure (114d) are also present in the three other structures, resulting in the wanted entailment. Note that *e* is the overall event, *t* is Jane, *p* is the path, *pl* is the place, and *t<sub>2</sub>* is the mountain.

- (114) a. [GO<sub>LOC</sub>(*e*) & ARG1(*e*, *t*) & JANE(*t*) & ARG2(*e*, *p*) & TO(*p*) &  
ARG1(*p*, *pl*) & TOP-OF(*pl*) & ARG1(*pl*, *t<sub>2</sub>*) & MOUNTAIN(*t<sub>2</sub>*)]  
b. [GO<sub>LOC</sub>(*e*) & ARG1(*e*, *t*) & JANE(*t*) & ARG2(*e*, *p*) & UP(*p*)]  
c. [GO<sub>LOC</sub>(*e*) & ARG1(*e*, *t*) & JANE(*t*) & ARG2(*e*, *p*) & DOWN(*p*)]  
d. [GO<sub>LOC</sub>(*e*) & ARG1(*e*, *t*) & JANE(*t*) & ARG2(*e*, *p*)]

However, because of the different path functions used in the first three structures (which are *TO*, *UP* and *DOWN*), structure (114a) does not entail structure (114b), as it should. In other words, the function describing the path in structure (114b) is not present in structure (114a). The fact that structure (114a) should not entail structure (114c) is captured, on the other hand, because of the same reason. This could be an indication that

Jackendoff wishes to imply that structure (112a) does not entail structure (112b) after all, as there might be a situation where Jane goes to the top of the mountain without actually climbing up the mountain.

But this seems to equate the meaning of sentence (112a) with that of *reaching the top of the mountain*, which is incorrect. To reach a mountain top you can, for instance, go on a locative path across a ladder from a helicopter directly to the top of the mountain. But to climb a mountain the path you go across must be on the mountain itself. And the top of the mountain can then only be reached if this path leads from somewhere and to the actual top, which can really only be captured by the preposition *up*. So, in a way, the way you climb a mountain is *by climbing up the mountain*.

If we follow this view, a solution in the conceptual structures might be to add a *BY* function, saying that Jane went to the top of the mountain (the structure in (113a)) by doing the same activity as in structure (113b). As a result, the structure for sentence (112a) would be the one shown in (115).

- (115) [event GO<sub>LOC</sub> ([thing JANE],  
           [path TO ([place TOP-OF ([thing MOUNTAIN]))])] ]  
       [BY [event GO<sub>LOC</sub> ([thing JANE],  
           [path UP ])] ] ]

If this structure is used, the wanted entailments are captured, because both structures (113b) and (113d) are embedded inside it, while structure (113c) is not. Even if this meaning is not what Jackendoff originally intended, this still illustrates that the conceptual theory can be pretty dynamic, as new and wanted semantic ideas can be added quite easily.

### 11.2.2 Parsons' Underlying Event Representation

Though Parsons does not include *climb* in any of his lists [25, p. 264-269], we can imagine that the sentences can be represented as underlying event formulas quite easily, as they are not causatives or inchoatives. For some proposed formulas, let's assume that Jane has the role Agent-Theme here, as she has both the characteristics of these two roles in these events. She is the one that moves, and she is also the one instigating this movement.

However, this raises the question of what role the mountain plays in these events. It seems reasonable that it would be the Goal, though that contradicts the fact that Jane is actually on the mountain during the event, not just at the end. Logically, climbing a mountain involves moving from the bottom of the mountain to the top of it. Therefore, the Source would be the bottom and the Goal would be the top, and the area between those two would be the path or trajectory. Since Parsons' theory does not include any thematic roles of the latter nature, we can assume the correct role for the mountain in this case is the Goal. We can also assume that the prepositional phrases are added to the formulas as predicates, as has been done earlier. As such, the formulas for the sentences in (112) are shown in formulas (116).

- (116) a.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Climbing}(e) \ \& \ \text{Agent-Theme}(e, \text{Jane}) \ \& \ \text{Goal}(e, \text{Mountain})]$

- b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Climbing}(e) \ \& \ \text{Agent-Theme}(e, \text{Jane}) \ \& \ \text{Up}(e, \text{Mountain})]$
- c.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Climbing}(e) \ \& \ \text{Agent-Theme}(e, \text{Jane}) \ \& \ \text{Down}(e, \text{Mountain})]$
- d.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Climbing}(e) \ \& \ \text{Agent-Theme}(e, \text{Jane})]$

All of these formulas are *Climbing* events that culminated in the past, where Jane was the Agent-Theme. Because formula (116d) is the same event as the other three, except without the predicates that they add, the formulas in (116a), (116b) and (116c) all correctly entail formula (116d). However, because the mountain appears in two different predicates in formulas (116a) and (116b), and none of these predicates are shared, formula (116a) does not entail formula (116b). For this to work, the *Up* predicate would need to be part of formula (116a), but that is not obvious from the surface sentence. Because of this, the entailment in question is not fully covered in Parsons' underlying event theory. However, the fact that formula (116a) does not entail formula (116c) is indeed captured, because of the different predicates.

### 11.2.3 VerbNet's Semantic Representation

The verb classes that have *climb* as one of their members include *calibratable\_cos-45.6-1*, *escape-51.1-3*, *meander-47.7* and *run-51.3.2*. We can exclude the class *calibratable\_cos-45.6-1* because it implies a meaning where something goes up and down on a scale, as well as the class *meander-47.7* because it involves something occupying the entirety of some location, exemplified by the sentence *the river climbed up the mountain*.

So, because the other two classes involve the motion of some Theme, this section will be considering the frames given by both of them. Note also that in both classes *climb* refers to the same sense in WordNet, namely "go upward with gradual or continuous progress", and to the same three frames in FrameNet [22]. Let's start by looking at the semantic structures given by the class *escape-51.1-3* for these sentences, shown in (117), where the *T* subscript signifies the Theme.

- (117)
- a.  $\text{MOTION}(\text{DURING}(E), \text{Jane}_T) \ \& \ \text{PATH}(\text{DURING}(E), \text{Jane}_T, ?\text{INITIAL\_LOCATION}, \text{Mountain}_{\text{TRAJECTORY}}, ?\text{DESTINATION})$
  - b.  $\text{MOTION}(\text{DURING}(E), \text{Jane}_T) \ \& \ \text{PATH}(\text{DURING}(E), \text{Jane}_T, ?\text{INITIAL\_LOCATION}, \text{Mountain}_{\text{TRAJECTORY}}, ?\text{DESTINATION})$
  - c.  $\text{MOTION}(\text{DURING}(E), \text{Jane}_T) \ \& \ \text{PATH}(\text{DURING}(E), \text{Jane}_T, ?\text{INITIAL\_LOCATION}, \text{Mountain}_{\text{TRAJECTORY}}, ?\text{DESTINATION})$
  - d.  $\text{MOTION}(\text{DURING}(E), \text{Jane}_T) \ \& \ \text{PATH}(\text{DURING}(E), \text{Jane}_T, ?\text{INITIAL\_LOCATION}, ?\text{TRAJECTORY}, ?\text{DESTINATION})$

In all four events, the predicates *MOTION* and *PATH* are used. The former says that Jane is in motion during the event, while the latter says that Jane also goes on a path during the event from some unknown Initial

location, through some Trajectory and to some Destination. The structures differ in whether or not they specify the trajectory, which is the mountain in structures (117a), (117b) and (117c).

Note, however, that there are three different frames available for the phrase structure *NP V PP*, which applies to the sentences in (112b) and (112c). The only difference between those three frames is the fact that the argument appearing inside the prepositional phrase (the mountain in this case) is either said to be either the Initial location, the Trajectory or the Destination. Assuming that none of the two sentences in question constitute the idea that Jane starts from the mountain or ends at the mountain, the most probable role is that of Trajectory. This notion is supported by the fact that the frame for the sentence in (112a) clearly specifies that the role of the mountain is, in that case at least, the Trajectory. Although the only thing the frames mentioned say about what the given preposition must be is that it must constitute a path, it seems natural to assume that an Initial location is given by the preposition *from*, while the Destination is given by *to*. So again, because the prepositions used in the example sentences are *up* and *down*, not *from* or *to*, the most logical role is that of Trajectory.

Because the structure in (117d) is the same as the other three, except with the Trajectory left unspecified, the structures in (117a), (117b) and (117c) correctly entail structure (117d). However, because structures (117a), (117b) and (117c) are all exactly the same, many wrong entailments exist here. For example, according to these structures there is no difference between climbing up a mountain and climbing down it. In addition, there is no difference between the transitive version, where the *mountain* is a direct object, and the prepositional phrase versions. There is also no representation of the fact that Jane reaches the top of the mountain in structure (117a).

So, let's examine the frames available in the class *run-51.3.2*. Note that this is the same superclass as the one used in the *jump* example (see section 11.1), so the semantic structures may seem quite familiar. However, there is no frame in this class that represents the phrase structure *NP V NP*, which applies to the sentence in (112a). Because of this, let's look at the frames available in the subclasses. One frame present, that does represent the given phrase structure, seems to be the most logical for the given sentence. The alternative structures for the sentences in question are shown in structures (118), where structure (118a) is taken from the subclass *run-51.3.2-1*.

- (118) a. MOTION(DURING(E), Jane<sub>T</sub>) &  
VIA(E, Jane<sub>T</sub>, Mountain<sub>LOCATION</sub>)  
b. MOTION(DURING(E), Jane<sub>T</sub>) &  
UP<sub>PREP</sub>(E, Jane<sub>T</sub>, Mountain<sub>LOCATION</sub>)  
c. MOTION(DURING(E), Jane<sub>T</sub>) &  
DOWN<sub>PREP</sub>(E, Jane<sub>T</sub>, Mountain<sub>LOCATION</sub>)  
d. MOTION(DURING(E), Jane<sub>T</sub>)



We see here that the *MOTION* predicate is again used, as it was used in the structures in (117). However, whereas those used the predicate *PATH*, the structures in (118) use the *PREP* predicate (also seen in the VerbNet structures in section 9.1)). Because of this, the different prepositions in the sentences result in different predicates, though they always take the same arguments. But again, there is no equality between the predicate in structures (118a) and (118b), which results in a lack of the wanted entailment. It would have been present if the sentence in (112b) was instead *Jane climbed via the mountain*, though that sentence sounds rather unnatural. Considering the fact that the semantic structure in (118a) isn't actually available for the verb *climb* (it comes from the subclass *run-51.3.2-1*), the entailment proposed is not captured in these structures. And because the same entailment wasn't captured in structures (117), the idea that sentence (112a) should entail sentence (112b) is not included in VerbNet, though the structures do capture the fact that sentences (112a), (112b) and (112c) all entail sentence (112d).

#### 11.2.4 Notable Differences and Similarities

Although Jane is appointed as the Agent-Theme in all of Parsons' structures, seen in formulas (116), she only has the role of Theme in Jackendoff's and VerbNet's structures, seen in structures (113) and (117), respectively. In the former structures, Jane acts as the first argument to the *GO<sub>LOC</sub>* function, while in the latter, she is directly given the role of Theme because of her position in the syntactic structure. In other words, she is the Theme because she is the subject in active *climb*-sentences.

The mountain, on the other hand, acts in a variety of different predicates and functions in Parsons' and Jackendoff's structures, though it is always the Trajectory in the VerbNet representations. Following Jackendoff [15, p. 258], the role of the mountain in structure (113a) is most probably that of Goal, because it acts as an argument to the path function *TO*. However, note that it is the place function *TOP-OF* that is the actual argument to the *TO* function, while the mountain is further the argument to *TOP-OF*. Because of this, it is the actual top of the mountain that is the Goal, not the whole mountain. Still, because the argument to a place function must be either the Source, Goal or Location, the role of the mountain seems to be that of Location, because only *FROM* and *TO* constitute the Source and Goal, respectively.

So, there are actually, in a sense, three roles in Jackendoff's structure, namely that of the Theme, the Location and Goal. In other words, the sentence in (112a) can be said to involve a Theme (Jane) going to a Goal (the top of the mountain), which is part of a Location (the mountain). This notion is missing in the other two representations, which does not include the fact that the top was reached in sentence (112a). If VerbNet did include it, though, then the structure in (117a) would say that the Destination is the *Top-Of-The-Mountain*, or something of that nature. Similarly, the underlying event formula would need to say that the Goal is the top of the mountain, while the area climbed is the whole of the mountain. The current structures

do not include this information, however.

Both the structures in (113) and (117) say that Jane goes on a path, which is specified by different functions in the conceptual structures, and refers to three different parts in the VerbNet structures, namely the Trajectory, the Initial location and the Destination. The latter two are said to be unknown in (117), but are not part of the structures at all in (113). If they were included, then the path in question would have included the *FROM* and *TO* functions as well, constituting the Source and the Goal, respectively. So, while VerbNet assumes that there is an understood, but unknown, Initial location, or Source, and similarly a Destination, or Goal, these are simply not included in the conceptual structures. This is similar to how it is done in Parsons' formulas, where the predicates for the Goal and the Source is not included, except for the inclusion of the mountain as the Goal in (116a). However, a more fitting role for the mountain in that case would be either Location or Trajectory, if we follow the structures given by Jackendoff and VerbNet. Although such a role is not provided in Parsons [25], it seems relatively easy to just add it.

One thing that the conceptual structures include, that the VerbNet structures lack, is the fact that the path is said to be locative. In structure (117), the nature of the path, or what kind of path it is, is left unspecified. While a path involving possession, for instance, would be represented in a conceptual structure by a *GO* event under the possessive semantic field, rather than the locative, this kind of path would instead involve a different predicate all together in VerbNet. This will become clearer in the next chapter (chapter 12), where the transferring of possessions will be discussed. However, where Jackendoff and VerbNet operate with a path that the Theme goes on, the underlying event formulas does not say anything about this. This is yet another illustration of the fact that the underlying event formulas do not go into detail as to what happens in a given event. They only include the name of the event, and the various arguments included in the surface sentence.

Similar to what was mentioned in section 11.1.4, the fact that Jane is in motion during the event is explicitly included in the VerbNet structures, with the use of the *MOTION* predicate. This fact is less clear in the other representations, though a *GO<sub>LOC</sub>* event does involve the locative movement of its first argument. Also, a Theme is said to be the thing being moved or changed (add Reference), so this fact is also present in Parsons' formulas, though not as direct as in VerbNet.

Although all three representations capture the fact that sentences (112a), (112b) and (112c) all entail sentence (112d), the idea that sentence (112a) should entail sentence (112b) is not really captured by any of them. The use of the alternative conceptual structure seen in (115) does solve this for Jackendoff's theory, though such a fix is harder in the other two representations. The entailment is indeed present in VerbNet, because the structures are all alike, but this results in the unnatural idea that climbing up a mountain is the same as climbing down it. If we had included the fact that the top of the mountain was reached in sentence (112), resulting in a sentence like "Jane climbed the mountain to the top of

the mountain", the semantic structure would, hypothetically, have included both the Trajectory and the Destination, and the structures would then not all entail one another. However, such a frame is unavailable in the *escape-51.1* class, where only one frame involves more than one part of the path (namely the Initial location and the Destination).

So, although the entailment between sentences (112a) and (112b) is captured in VerbNet, many other entailments are also incorrectly included. The fact that the wanted entailment is not included in Jackendoff's or Parsons' structures indicates that the two authors believe there is no entailment relation present between the two sentences (though they both entail the sentence in (112d)). However, arguments provided in this chapter, particularly in the section on Jackendoff's structures, illustrate that such an entailment is, at least, worth consideration.



## Chapter 12

# Verbs of Change of Possession

Situations described by these kind of verbs involve the exchange of something between entities, typically a giver or seller and a receiver or buyer. Example verbs are *find*, *loan* and *trade*, though this chapter will consider perhaps the most talked about kind of situation, namely one described by the verbs *buy*, *sell* and *pay*. Though these are three quite different verbs on the surface, they do seem to be strongly related [17, p. 138].

### 12.1 Buy/Sell/Pay

*Buy*, *sell* and *pay* are verbs that seem to offer different perspectives on the same situation. This situation, which can be called a purchase, typically involves a buyer, a seller, the thing being bought and the amount of money used in the transaction. For instance, FrameNet proposes three frames which all involve some sort of commerce, called *Commerce\_buy*, which has *buy* as a lexical unit member, *Commerce\_sell*, which has *sell* as a member, and *Commerce\_pay*, which has *pay* as a member. In all three frames, the roles *Buyer*, *Seller*, *Goods* and *Money* are available, although there is a difference between the frames regarding whether these are core or non-core roles [1]. Consider the example sentences in (119) as an illustration of how a transaction can be referred to.

- (119) a. *Keith bought the car from Joel for money.*  
b. *Joel sold the car to Keith for money.*  
c. *Keith payed Joel money for the car*

Here, we can assume that *Keith* is the buyer, *Joel* is the seller, *the car* is the thing being bought or sold, and *money* is the amount used in the transaction. Likewise, we might further assume that *Keith* is the Agent of the buying event, *Joel* is the Agent of the selling event, *the car* is the Theme in all three sentences, and *money* has the same role as well in all three, perhaps as an Asset or Amount. Finally, one could state that the three sentences all describe the same event, but viewed from different perspectives.

However, Parsons claims that these verbs do in fact describe different events [25, p. 84]. If the event is the same, *Joel* would both be the Agent of

the selling and the buying, and *Keith*, likewise, would be the Agent of both the buying and the selling. Another relevant argument to this, according to Parsons, is the fact that you could add an Instrument to the buying event, such as a credit card, and it would, in most cases, be false to say that that Instrument also applies to the selling event. Even though Keith might buy the car with a credit card, Joel would not sell it with it. Joel might have used some other Instrument instead, such as a sales pitch, for instance. In other words, a transaction would typically involve more than one event (a buying, selling and paying event in this case).

When it comes to the notion of entailment here, it seems logical to think that all three sentences entail one another. If there is a buying event, then there must also be selling and paying events, and similarly the other ways around. As such, let's assume that sentence (119a) entails sentences (119b) and (119c), sentence (119b) entails sentences (119a) and (119c), and sentence (119c) entails sentences (119a) and (119b).

In a way, these verbs are also related to the verb *own*, as a buying or a paying event would result in the fact that the buyer or payer owns the thing he received in the transaction (at least for some time). Likewise, a seller would not own the thing anymore after a selling event. Therefore, let's also assume that the sentences in (119a) and (119c) both entail the sentence in (120a), and the sentence in (119b) entails the sentence in (120b). For this, we must imagine that *the car* represents a particular car in this case, as it is perfectly possible that Joel might own some other car.

- (120) a. *Keith owns the car.*  
       b. *Joel does not own the car.*

In other words, the events that are represented by the sentences in (119) bring about a state where Keith owns the car, which is represented in sentence (120a). They also bring about a state where Joel does not own the car in question anymore (the state in sentence (120b)).

### 12.1.1 Jackendoff's Conceptual Structure Representation

Jackendoff [15, p. 191] lists lexical entries for the verbs in question. All three make use of the functions *CAUSE*, *GO<sub>POSS</sub>* and *EXCH*, something which signifies that Jackendoff wishes to express their apparent similarity. Actually, all three structures are exactly the same, except for the fact that the arguments to each function are switched around. The conceptual structures for the sentences in (119) are shown in structures (121) below.

- (121) a. [event CAUSE ([thing KEITH], [event GO<sub>POSS</sub> ([thing CAR],  
                                   [path FROM ([thing JOEL])  
                                   TO ([thing KEITH])])  
                                   [EXCH [event GO<sub>POSS</sub> ([thing MONEY],  
                                   [path FROM ([thing KEITH])  
                                   TO ([thing JOEL])])])]]]  
       b. [event CAUSE ([thing JOEL], [event GO<sub>POSS</sub> ([thing CAR],

- [path FROM ([thing JOEL])  
TO ([thing KEITH]))]  
[EXCH [event GO<sub>POSS</sub> ([thing MONEY],  
[path FROM ([thing KEITH])  
TO ([thing JOEL]))]]]
- c. [event CAUSE ([thing KEITH], [event GO<sub>POSS</sub> ([thing MONEY],  
[path FROM ([thing KEITH])  
TO ([thing JOEL]))])]  
[EXCH [event GO<sub>POSS</sub> ([thing CAR],  
[path FROM ([thing JOEL])  
TO ([thing KEITH]))]]]

As can be seen, the only difference between the three structures are the arguments to the *GO<sub>POSS</sub>* functions, both the one used as the effect of the main *CAUSE* function, and the one used in the *EXCH* function. Remember that *EXCH* signifies something that is done in exchange for some other action. In (121a) and (121b) this exchange involves the transfer of *MONEY* from *KEITH* to *JOEL*, while in (121c) the thing being transferred is rather the *CAR*, now moving from *JOEL* to *KEITH*. In both (121a) and (121b), the thing being used as the first argument to the *GO<sub>POSS</sub>* function inside *CAUSE* is the car, and it moves, logically, from Keith to Joel. In (121c), on the other hand, this thing is instead the money, and it moves from Joel to Keith.

Although Jackendoff doesn't present a lexical entry for the verb *own*, one can imagine that it also involves the possessive semantic field. However, since the sentences in (120) represent states, not events (as the sentences in (119) do), the proper analysis would probably be to use the inchoative function here. As Jackendoff [15, p. 93-94] argues, some *GO* events can be reduced to *INCH* events if they take a *BE* function as its argument (note that there are a few cases where this would be not be possible). Assuming that this redundancy is applicable to the events in question, the sentences in (120) would therefore have structures that simply utilize the *BE<sub>POSS</sub>* function, without the surrounding *INCH* function. In summary, the *GO<sub>POSS</sub>* events in structures (121) could have alternative forms using *INCH* and *BE<sub>POSS</sub>*, and the structures representing *own* would only be the *BE<sub>POSS</sub>* structure used in those. The structures in (122) show the resulting conceptual structures. Notice the use of the *NOT* function, which indicates negation, representing the state that Joel does not possess (or own) the car.

- (122) a. [state *BE<sub>POSS</sub>* ([thing CAR], [place AT ([thing KEITH]))])  
b. [state NOT *BE<sub>POSS</sub>* ([thing CAR], [place AT ([thing JOEL]))])]

Although it seems fair to suggest that structure (121a) and structure (121b) entail one another because they are almost the same structure, there is a difference when it comes to who the causer (or Agent) is. In structure (121a) it is Keith that is the Agent, but in structure (121b) it is Joel. In structure (121c) the two *GO<sub>POSS</sub>* events are also present in the other two structures, except that their positions have been swapped; structures (121a)

and (121b) use one event as the effect and the other inside the *EXCH* function, but in structure (121c), it is the other way around. Aside from that, the Agent is again Keith, a difference from structure (121b).

So, because of these differences between the structures, it seems logical to conclude that the wanted entailments are not captured here. However, these observations may be an indication that the entailments proposed earlier were in fact false, as Jackendoff clearly aims to make these three events distinct. Before adopting this point of view, however, the next two sections will examine the representations employed by Parsons and VerbNet.

First, though, let's examine the *own* structures. It seems logical to suggest that the three structures in (121) all entail the structure in (122a), if we assume that the *GO<sub>POSS</sub>* function can have the alternative structure mentioned. As such, the correct entailments are, indeed, captured here. The notion that structure (122b) is also entailed is a little less clear, on the other hand. The fact that the possessive state should be negated doesn't seem to be encoded within the framework. If it was, then it would be clear that a *GO<sub>POSS</sub>* event that uses *FROM* as one of its path functions should entail a state where the thing being own does not belong to the thing used in the *FROM* function anymore. However, Jackendoff does not say anything about this issue, so one can assume that this particular entailment is also not captured in the current framework.

### 12.1.2 Parsons' Underlying Event Representation

Parsons lists the three verbs in question as ditransitive verbs involving both a direct and indirect object [25, p. 266]. For *buy*, the indirect object has the role of Benefactive, while for *sell* and *pay* the indirect object is instead a Goal. *Own*, on the other hand, is listed as a transitive verb, where the subject is an Experiencer [25, p. 266]. Utilizing the templates Parsons provides, the resulting formulas for the three sentences in (119) are shown in (123), while the formulas for the sentences in (120) are shown in (124). Note the use of the negation symbol in formula (124b) which indicates that there is no state where Joel owns the car in question.

- (123) a.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Buying}(e) \ \& \ \text{Theme}(e, \text{Car}) \ \& \ \text{Agent}(e, \text{Keith}) \ \& \ \text{From}(e, \text{Joel}) \ \& \ \text{For}(e, \text{Money})]$
- b.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Selling}(e) \ \& \ \text{Theme}(e, \text{Car}) \ \& \ \text{Agent}(e, \text{Joel}) \ \& \ \text{Goal}(e, \text{Keith}) \ \& \ \text{For}(e, \text{Money})]$
- c.  $(\exists e)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(e, t) \ \& \ \text{Paying}(e) \ \& \ \text{Theme}(e, \text{Money}) \ \& \ \text{Agent}(e, \text{Keith}) \ \& \ \text{Goal}(e, \text{Joel}) \ \& \ \text{For}(e, \text{Car})]$
- (124) a.  $(\exists s)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(s, t) \ \& \ \text{Owning}(s) \ \& \ \text{Theme}(s, \text{Car}) \ \& \ \text{Experiencer}(s, \text{Keith})]$



- b.  $\neg(\exists s)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Cul}(s, t) \ \& \ \text{Owning}(s) \ \& \ \text{Theme}(s, \text{Car}) \ \& \ \text{Experiencer}(s, \text{Joel})]$

Because the events are all of different kinds in formulas (123) (*Buying*, *Selling*, and *Paying* events, respectively), they do not entail one another. They also use different thematic roles; the car is the Theme in formulas (123a) and (123b), but in formula (123c) the money is instead the Theme, for instance. The Agents also differ between formula (123b) and the other two. So, similar to Jackendoff, Parsons maintains that there are no entailments present between the three sentences. This therefore illustrates further evidence of the idea that the entailments proposed at the start of this chapter were not correct.

Similarly, there is no relationship between any of the formulas in (123) and (124), mostly because the latter are *Owning* events. A possible solution to this might be to use a meaning postulate which says that if there exists either one of the events in formulas (123), then there also exists an owning state where the Theme is the same, and the Experiencer depends on the given event. However, Parsons has not provided any such postulate, which leads to the conclusion that he does not agree with the entailments proposed.

### 12.1.3 VerbNet's Semantic Representation

Finding the correct semantic structures for the four verbs in VerbNet proved to be quite easy, as each one is only a member of a single class. *Buy* is a member of the class *get-13.5.1*, *sell* is a member of *give-13.1-1*, *pay* is a member of *pay-68-1*, and *own* is a member of *own-100*. Given the frames available in these classes, the structures for the sentences in (119) are shown in (125), while the structures for the sentences in (120) are shown in (126). All classes, except *own-100* uses a thematic role called Asset, which applies to the money used. This role is indicated by the subscript *AS*. The roles of Theme and Agent are also used in all the classes. However, while *give-13.1-1* and *pay-68-1* make use of a Recipient role, *get-13.5.1* lacks this and instead utilizes a Source and a Beneficiary, both of which are not present in the other classes. The Recipient is indicated by the *R* subscript, and the Source by the *S* subscript (the Beneficiary is not used here). As before, the *A* subscript constitutes the Agent, while *T* constitutes the Theme. Also, the *HP* predicate is an abbreviation of the predicate *HAS\_POSSESSION*.

- (125) a.  $\text{HP}(\text{START}(E), \text{Joel}_S, \text{Car}_T) \ \& \ \text{HP}(\text{END}(E), \text{Keith}_A, \text{Car}_T) \ \& \ \text{TRANSFER}(\text{DURING}(E), \text{Car}_T) \ \& \ \text{COST}(E, \text{Money}_{AS})$   
 b.  $\text{HP}(\text{START}(E), \text{Joel}_A, \text{Car}_T) \ \& \ \text{HP}(\text{END}(E), \text{Keith}_R, \text{Car}_T) \ \& \ \text{HP}(\text{START}(E), \text{Keith}_R, \text{Money}_{AS}) \ \& \ \text{HP}(\text{END}(E), \text{Joel}_A, \text{Money}_{AS}) \ \& \ \text{TRANSFER}(\text{DURING}(E), \text{Car}_T)$   
 c.  $\text{TRANSFER}(\text{DURING}(E), \text{Car}_T) \ \& \ \text{TRANSFER}(\text{DURING}(E), \text{Money}_{AS}) \ \& \ \text{HP}(\text{START}(E), \text{Keith}_A, \text{Money}_{AS}) \ \&$

NOT(HP(START(E), Keith<sub>A</sub>, Car<sub>T</sub>)) & HP(END(E), Keith<sub>A</sub>, Car<sub>T</sub>) & NOT(HP(END(E), Keith<sub>A</sub>, Money<sub>AS</sub>)) &  
VALUE(E, Car<sub>T</sub>, Money<sub>AS</sub>)

Note that the *NOT* predicate has been added in to the structure in (126b), in order to represent the negation.

- (126) a. HAS\_POSSESSION(E, Keith<sub>PIVOT</sub>, Car<sub>T</sub>)  
b. NOT(HAS\_POSSESSION(E, Joel<sub>PIVOT</sub>, Car<sub>T</sub>))

All of these semantic structures use the predicate *HAS\_POSSESSION*, something which indicates their similarity. Each structure in (125) says that Keith possesses the car at the end of the event, though the role he occupies differ. In (125a) and (125c), Keith is the Agent, but in (125b), he is instead the Recipient. Also, in all three structures in (125), the car is transferred during the event, using the predicate *TRANSFER*.

But other than those similarities there are many differences as well. Structures (125b) and (125c) both represent the fact that the money is also transferred in the event, from Joel at the start to Keith at the end, by use of the *HAS\_POSSESSION* predicate. Structure (125a), on the other hand, doesn't say anything about this. However, structure (125c) is the only one that actually specifies that the money is transferred during the event (using the *TRANSFER* predicate). Structure (125c) also specifies that Keith does not possess the car at the start of the event, and that he does not possess the money at the end of it. In addition, structure (125a) uses a predicate called *COST*, which seems to say that the event costs money, while structure (125c) uses a predicate called *VALUE*, which says that the car has the value given by the money. Finally, an interesting observation is that the Recipient role in the semantic structure for *pay* (which is Joel in this case) is not used at all, even though it is part of the syntactic structure.

The *own* sentences, on the other hand, are much simpler, and they also make use of the *HAS\_POSSESSION* predicate. Because they are states, the entire *E* variable is used as an argument to the predicate. The role applied to the subject is different from the structures in (125), however, being instead something called *Pivot*.

Because of the mentioned differences between the structures in (125) they do not entail one another, a notion that we should now assume to be correct, considering the fact that Jackendoff and Parsons also agreed on it. If we ignore the different labeling of participants, then there do exist entailment relations to the sentence in (126a), however. All three structures in (125) entail structure (126a), if we assume this, because they all include the fact that Keith has possession of the car at the end of the event. Structure (126b), on the other hand, is not entailed by any of the structures, though this would have been the case if a predicate like *NOT(HAS\_POSSESSION(END(E), Joel<sub>S</sub>, Car<sub>T</sub>))* was used, for instance.

#### 12.1.4 Notable Differences and Similarities

Although there are differences between the various structures as to what roles the participants occupy, the three representations at least agree that Keith is the Agent in both sentences (119a) and (119c), and that Joel is the Agent in sentence (119b). They also agree that Keith is the Goal or Recipient in sentence (119b), while Joel is the Source in sentence (119a) and the Goal in sentence (119c). Remember that the Source and Goal are constituted by the first arguments to the functions *FROM* and *TO*, respectively, in the conceptual structures, while the Agent is the first argument to the *CAUSE* function. Although Parsons uses the predicate *From*, rather than *Source*, we can assume that they mean the same thing.

Jackendoff and VerbNet agree that the car is the Theme in all the three sentences, but while Parsons labels the car as the Theme in formulas (123a) and (123b), this participant is instead used as an argument to the predicate *For* in formula (123c). While VerbNet labels the money as the Asset of the event, Jackendoff instead says that it is the Theme, because it acts as an argument to a *GO* function. This illustrates the fact many participants can occupy the same role in an event in Jackendoff's theory, as both the car and the money are Themes in structures (121). In other words, they are both arguments to *GO* functions, one inside the *CAUSE* event, and the other inside the *EXCH* event. However, when the arguments appear inside the *EXCH* function, an alternative thematic role name is perhaps more suitable. For comparison with the other representations, let's instead call it the Asset.

So, the roles that the car and the money occupy are consistent throughout Jackendoff's and VerbNet's structures, while they differ somewhat in Parsons' formulas. The money is used as an argument to the *For* predicate in formulas (123a) and (123b), but is labeled as the Theme in formula (123c). In other words, if we imagine that the *For* predicate constitute the role of Asset (in order to compare to VerbNet), Parsons and VerbNet agree that the money is the Asset in sentences (119a) and (119b), and that the car is the Theme in these same sentences, but they disagree in their structures for sentence (119c). In the latter, Parsons says that the car is the Asset and the money is the Theme, while VerbNet says the opposite; that the car is the Theme and the money the Asset.

In addition, Keith and Joel actually occupy up to three different roles in some of the conceptual structures, because they appear more than once, as arguments to different functions. This means that Keith is both the Agent, Goal and Source in structures (121a) and (121c), and both the Source and Goal in structure (121b). Similarly, Joel is both the Agent, Source and Goal in structure (121b), and the Source and Goal in structures (121a) and (121c). This illustrates the fact that participants can also occupy many different roles in Jackendoff's theory, a notion that is more or less missing in the other two theories, though Parsons does include conjuncted thematic roles (like *Agent-Theme* etc.). As a final illustration of this runthrough of the various thematic roles, take a look at the tables below, which show what roles the participants occupy in each of the three structures given by the three theories. Note that *A* is the Agent, *T* is the Theme, *S* is the Source, *G*

is the Goal, and *AS* is the Asset.

Jackendoff	a.	b.	c.	Parsons	a.	b.	c.
KEITH:	A,S,G	S,G	A,S,G	KEITH:	A	G	A
JOEL:	S,G	A,S,G	S,G	JOEL:	S	A	G
CAR:	T	T	AS	CAR:	T	T	AS
MONEY:	AS	AS	T	MONEY:	AS	AS	T

VerbNet	a.	b.	c.
KEITH:	A	G	A
JOEL:	S	A	G
CAR:	T	T	T
MONEY:	AS	AS	AS

So, while Keith and Joel are given specific roles in Parsons' and VerbNet's structures, based on the verb event given, they are always both the Source and the Goal in Jackendoff's structures. Jackendoff does differentiate between the events when it comes to what the Agent is, however. But while it is perfectly clear that the car or the money goes on a possessive path between Keith and Joel in all the three conceptual structures, the underlying event formulas only focus on either the Source or the Goal, plus the Agent and the Theme. This also applies to the VerbNet structures, which does not mention the fact that the Agent in each event is also either the Goal or Source. However, the idea of a Goal and a Source is still conveyed in VerbNet's structures, by the use of the *HAS\_POSSESSION* predicate. Though, as mentioned, the VerbNet structures differ in their inclusion of all the links in the transaction, as Joel is not included at all in structure (125c), for instance. Parsons, on the other hand, does not in any way include the fact that the Agent is also either the Goal or the Source.

While Jackendoff seems to focus on the fact that the three verbs in question are very much related, as evidenced by the use of almost the exact same structures for each of them, Parsons maintains that the events represented by the sentences in (119) should be considered separate. VerbNet does, similar to the structures provided by Jackendoff, focus on the events' similarity, evidenced, for instance, by the use of most of the same predicates, and the same thematic role labels. None of the representations maintain that there is any entailment relations between the three sentences, however. Still, there is a difference between sentences describing the same situation, and sentences that all have some meaning in common. In other words, while the sentences in (119) may not represent the exact same situation, or event, because of the different perspectives, there are still many similarities between them. This is at least the case in Jackendoff's and VerbNet's structures, which, as mentioned, all say that the car moves to Keith, for instance.

When it comes to the structures for the *own* sentences, all the representations agree that the car is the Theme throughout, but disagree on what the owner (Keith or Joel) should be called. Jackendoff puts the owner as an argument to the *AT* function, effectively making it the Location,

Parsons calls it the Experiencer, while VerbNet calls it the Pivot. Whatever label used, the name is at least consistent in the two structures given by each theory. The notion of negation is present in all three theories, so the sentence in (120b) is easily represented. However, as mentioned, none of the structures capture the fact that the sentences in (119) should entail sentence (120b), though they do, at least, capture the fact that they entail sentence (120a).



## Chapter 13

# Verbs of Change of State

These verbs describe an entity going from one kind of state to another, though only one of those states might be represented by a sentence employing a change of state verb. For instance, it might seem logical to assume that for a door to close it must be open first, though this might not always be the case, as a door might also be created to be closed from the start. In that case, it is true to say that the door became closed when it was made, though it would be wrong to say that it was open before that. A verb similar to *close* is *open*, which is the verb that will be examined in this entailment example. Other change of state verbs include *clean*, *darken* and *warm*, among many others [17, p. 240].

### 13.1 Open

The verb *open* can be used in both a transitive and an intransitive construction. When the transitive form is used, the subject of the sentence is whatever entity or force that causes the direct object to open. As such, there would be two participants, where the participant represented by the direct object would have its state changed to being open. In the intransitive variant, on the other hand, the subject is replaced by the direct object of the transitive sentence, though the original subject, which is the causing entity, might still be a part of the sentence, typically by appearing in a phrase following the preposition *by*. This resulting sentence would then be the passive alternative to the active transitive sentence.

If we assume that the passive and the active sentences both describe the same kind of event (except with different aspects), both of them should entail the intransitive sentence, where the causing entity is removed. Consider the examples shown in (127), where sentence (127a) is the active transitive, sentence (127b) is the passive version, sentence (127c) is the intransitive, and sentence (127d) represents the state of the door being open. Essentially, *Mary* does something that results in the door opening, which leads to the state of the door being open.

- (127) a. *Mary opened the door.*  
b. *The door was opened by Mary.*

- c. *The door opened.*
- d. *The door is open.*

Because this is an example of a causative-inchoative situation (as we have seen before, for instance in section 8.1), we want a semantic representation of these sentences to acknowledge the fact that both sentences (127a), (127b) and (127c) entail sentence (127d), while sentences (127a) and (127b) also entail sentence (127c). In other words, if Mary opened the door, as is done in sentences (127a) and (127b), then the door also opened, as is the case in sentence (127c), and following these events the door also ends up in the state of being open, as in sentence (127d). Note that WordNet lists the transitive and the intransitive versions of *open* as two separate senses, though they do share a cause relation. This is even evident in the name of the synset that the transitive variant is a member of, which is defined as "cause to open or to become open", while the intransitive sense is simply described as "become open" [18].

### 13.1.1 Jackendoff's Conceptual Structure Representation

Although *open* is thought of as a causative-inchoative verb, Jackendoff does not make use of the *INCH* function for his lexical entry of *open* [15, p. 252]. Instead, the function *GO<sub>IDENT</sub>* is used. Thus, the conceptual structures for the sentences in (127) are shown in (128). Note that structure (128a/b) represents both the sentence in (127a) and the one in (127b), because the conceptual theory does not make a distinction between passive and active forms.

- (128) a./b. [event CAUSE ([thing MARY],  
[event GO<sub>IDENT</sub> ([thing DOOR],  
[path TO ([place AT ([property OPEN])])])])])]  
c. [event GO<sub>IDENT</sub> ([thing DOOR],  
[path TO ([place AT ([property OPEN])])])]  
d. [state BE<sub>IDENT</sub> ([thing DOOR],  
[place AT ([property OPEN])])]

In both structures (128a/b) and (128c) the door goes on an identificational path to have the property *OPEN*. The difference between them is that structure (128a/b) has a surrounding *CAUSE* function which says that Mary is the Agent causing the *GO<sub>IDENT</sub>* event. However, in neither structure (128a/b) nor structure (128c) is the state of the door, which is represented in structure (128d), a part of the structure. This is because the *GO* function is used instead of *INCH* in the former structures. However, because Jackendoff assumes that the kind of *GO* event in question can be reduced to a structure involving the functions *INCH* and *BE*, the events in structures (128a/b) and (128c) do indeed result in a state where the door is said to have the property of being open. This is essentially the same kind of situation as we saw for the *buy/sell/pay* example (see section 12.1.1), where



a possessive *GO* event was reduced to a possessive *INCH* event, taking a *BE* function as its argument.

So, given that the shared *GO<sub>IDENT</sub>* event that occurs within structures (128a/b) and (128c) can also have the alternative structure seen in (129), the wanted entailments are captured. In other words, the state in structure (128d) is also embedded within structures (128a/b) and (128c) with this alternative structure, so the latter two correctly entail structure (128d). In addition, structure (128a/b) entails structure (128c) because the latter is embedded inside the former. This might become more apparent if we take a look at the alternative flat structures, shown in (130), where all the functions that are present in structure (130d) are also present in structure (130c), and all the functions in structure (130c) are also present in structure (130a/b). Note that flat structures make use of the *INCH* and *BE* functions, rather than the *GO* function. Note also that the variable *s* is the *BE<sub>IDENT</sub>* state, *p* is the place, *p<sub>2</sub>* is the open property, *t* is the door, *t<sub>2</sub>* is Mary, *e* is the inchoative event, and *e<sub>2</sub>* is the causative event.

(129) [event INCH ([state BE<sub>IDENT</sub> ([thing DOOR],  
[place AT ([property OPEN])]])]]

(130) a./b. [BE<sub>IDENT</sub>(*s*) & ARG1(*s*, *t*) & ARG2(*s*, *p*) & AT(*p*) &  
ARG1(*p*, *p<sub>2</sub>*) & DOOR(*t*) & OPEN(*p<sub>2</sub>*) & INCH(*e*) & ARG1(*e*, *s*)  
& CAUSE(*e<sub>2</sub>*) & ARG1(*e<sub>2</sub>*, *t<sub>2</sub>*) & ARG2(*e<sub>2</sub>*, *e*) & MARY(*t<sub>2</sub>*)]  
c. [BE<sub>IDENT</sub>(*s*) & ARG1(*s*, *t*) & ARG2(*s*, *p*) & AT(*p*) &  
ARG1(*p*, *p<sub>2</sub>*) & DOOR(*t*) & OPEN(*p<sub>2</sub>*) & INCH(*e*) & ARG1(*e*, *s*)]  
d. [BE<sub>IDENT</sub>(*s*) & ARG1(*s*, *t*) & ARG2(*s*, *p*) & AT(*p*) &  
ARG1(*p*, *p<sub>2</sub>*) & DOOR(*t*) & OPEN(*p<sub>2</sub>*)]

### 13.1.2 Parsons' Underlying Event Representation

This thesis has already touched upon a very similar verb to *open*, namely *close*, with regards to the underlying event theory (see sections 4.5 and 4.7). Even though Parsons lists passive templates separate from the active versions, he still proposes that their structures remain the same as their active counterparts [25, p. 272-273]. In other words, as was the case with the conceptual structures, there is no immediate difference in meaning between the active sentence in (127a) and the passive in (127b). Therefore, the formula shown in (131a/b) represents both the active and the passive sentences, while the other two represent the intransitive and the state sentence, respectively. These formulas are based on Parsons' template for causative-inchoative sentences.

(131) a./b. (∃e)(∃t)(∃I) [t ∈ I & I < now & Cul(e, t) & Agent(e, Mary) &  
(∃e<sub>2</sub>)(∃t<sub>2</sub>) [t<sub>2</sub> ∈ I & BECOME#(Open)(e<sub>2</sub>) & Theme(e<sub>2</sub>, Door) &  
Cul(e<sub>2</sub>, t<sub>2</sub>) & CAUSE(e, e<sub>2</sub>)]]  
c. (∃e)(∃t)(∃I) [t ∈ I & I < now & BECOME#(Open)(e) &  
Theme(e, Door) & Cul(e, t)]

- d.  $(\exists s)(\exists t)(\exists I) [t \in I \ \& \ I < \text{now} \ \& \ \text{Being-Open}(s) \ \& \ \text{Theme}(e, \text{Door})$   
 $\ \& \ \text{Hold}(e, t)]$

Similar to the treatment of *close*, the causative-inchoative formula in (131a/b) consists of two events, where one is embedded within the scope of the other. The surrounding causing event,  $e$  (where Mary is the Agent), is said to cause the other event,  $e_2$  (where the door is the Theme), which is an inchoative because of its use of the function *BECOME#*. This latter event results in the existence of a state of being open, because of the proposed meaning postulate discussed in section 4.5. This state is then the same formula as the one in (131d). Because formula (131c) is part of the formula in (131a/b), the latter entails the former, and because the state in formula (131d) is the same as the state that results from the inchoative event in the other two formulas, formulas (131a/b) and (131c) both entail formula (131d).

### 13.1.3 VerbNet's Semantic Representation

In the VerbNet database, the verb *open* appears as a member in five classes. However, only three of them seem to be related to the sense in question, since they are the only ones that refer to the mentioned sense in WordNet described as "cause to open or to become open" [18]. These classes are *crane-40.3.2*, *spatial\_configuration-47.6* and *other\_cos-45.4*. However, only *other\_cos-45.4* refers to the second relevant sense in WordNet as well, namely the one defined as "become open."

Although *spatial\_configuration-47.6* includes a frame for the intransitive phrase structure, it does not have a phrase structure that corresponds to the one in (127a), which is *NP V NP*. *Crane-40.3.2* does not include a frame for the intransitive sentence, and it also uses semantic predicates referring to the transferring of information, a meaning that does not seem to apply here (in addition, it also needs the Patient to be a body part). Therefore, *other\_cos-45.4* seems to be the most fitting class here, because it refers to the state of the Patient in its semantic structures.

So, given the frames for transitive and intransitive sentences in the class *other\_cos-45.4*, the structure for the sentences in (127a), (127b) and (127c) are shown in (132a/b) and (132c), respectively. For the structure in (132d), which represents the sentence in (127d), the frame used comes from the class *seem-109-1-1*. This is the only class that has *be* as its member, which does refer to the copular sense of the verb in WordNet. In other words, this frame is the only one available that represents simple state sentences, using a copula, though the semantic predicate used seems a little strange.

- (132) a./b.  $\text{CAUSE}(\text{Mary}_A, E) \ \& \ \text{STATE}(\text{RESULT}(E), \text{Open}_{\text{ENDSTATE}}, \text{Door}_P)$   
c.  $\text{STATE}(\text{RESULT}(E), \text{Open}_{\text{ENDSTATE}}, \text{Door}_P)$   
d.  $\text{SEEM}(E, \text{Door}_T, \text{Open}_{\text{ATTRIBUTE}})$

Note the use of the verb specific arguments *ENDSTATE* and *ATTRIBUTE* in these structures, where the former refers to the entire state of the

Patient, while the latter refers to only a single attribute of the Theme. In structure (132a/b), Mary is the cause of the whole event, which involves the state of the Patient (the door) ending up as being open, as a result of the event. In (132c), the *CAUSE* predicate has been removed, and the door simply ends up in a state of being open, as a result of the event. The structure in (132d), on the other hand, uses a different predicate (namely *SEEM*) and it also calls it the attribute of being open, not the endstate. Therefore, it is not entailed by the other two structures. An alternative to this would be to use the structure in structure (133), which uses the same *STATE* predicate as in structures (132a/b) and (132c).

(133)  $STATE(E, Open_{ENDSTATE}, Door_P)$

This structure constitutes the state of the door being open. If we assume that this is the correct structure for the sentence in (127d), then all the proposed entailments are captured in VerbNet. In other words, the *STATE* predicate used in structures (132), using the result stage of the event as an argument, brings about the *STATE* predicate in structure (133), where the whole of the event is used as the argument. However, because no frame is available for *be* which gives this specific structure, then there still seems like some connection here is missing.

#### 13.1.4 Notable Differences and Similarities

In all three representations, Mary is said to be the Agent (when she is part of the given sentences), and the door is said to be either the Theme or the Patient. While Jackendoff refers to the idea of being open as a property, Parsons uses the adjective in his *BECOME#* function, which constitutes the inchoative event where the Theme becomes open. VerbNet refers to *open* as an Endstate in structures (132a/b) and (132c), and as an Attribute in structure (132d), though the latter is, as mentioned, perhaps not the proper structure to use here.

The conceptual structure in (128a/b) refers to one overall causative event, involving another embedded inchoative event. This latter event takes a state as an argument, which involves the door having the property of being open. Similar to those structures, the underlying event formulas in (131) also refer to two events (the causative and the inchoative) and one state, where the inchoative event is embedded within the scope of the causative event. The state is, on the other hand, not directly part of the formulas in (131a/b) and (131c), but comes about because of the presumed meaning postulate. VerbNet, in contrast, refers only to one event here, and to the fact that the door is in the state of being open as a result of the event. The cause of the event is included, however, and the fact that a state is in question is conveyed by both the *STATE* predicate and the Endstate role. So, while Jackendoff and Parsons both refer to two events and one state, VerbNet only refers to one event and the state, not directly mentioning any inchoative.

Though the three structures appear very different from each other, the way the entailment relations work is actually quite similar in all

of them, at least the fact that sentence (127a) entails sentence (127c). In each representation, the latter structure is either embedded or a part of the former, resulting in the entailment. In structure (128c), the inchoative sentence's structure is the same as the argument used in the causative sentence's structure. Similarly, the underlying event formula for sentence (127c) is embedded inside formulas (131a/b). Though there is no embedding in the VerbNet structures, the structure in (132c) is also present in sentence (132a/b). The entailment relation to sentence (127d) is captured by Jackendoff because of the same embedding nature as in the other entailment, while a meaning postulate is used by Parsons. The capturing of this entailment is not present in VerbNet, however, if we assume the structure used in (132d). Still, the inclusion of the alternative structure proposed in (133) would capture it.

## Chapter 14

# Strengths and Weaknesses

After examining a number of interesting entailment phenomena and how the three theories of semantic representation handle each of them, this chapter will offer a conclusive overview of the various strengths and weaknesses of the three frameworks. All of them have had cases where a wanted entailment is not captured, either deliberately, where the entailment in question may have been thought of as non-existing by the author, or not.

In the latter case, the entailment is genuinely not included in the structure provided. However, it should be noted that although the entailment relation to the sentences given may not be present in one of the frameworks, other, weaker entailments might still be there. For instance, underlying event formulas for the *hit*-sentences (see structure (88)) always involve a *Hitting* event, so they all entail the quite simple sentence *There was a hitting*. In that case, there is a correct entailment present, even though the more complex entailments mentioned in that example were not captured.

Considering the detail of the various examples, this chapter will rather look at broader characteristics of the three frameworks, and how the mechanisms of each aim to account for and capture the notion of entailment. This will be done by going through some key points that are important in a semantic representation, most importantly with regards to entailment. The chapter will be discussing how the three theories deals with each specific issue, which will point out their strengths and weaknesses.

### 14.1 Coverage

Firstly, the notion of coverage, and the final listing of lexical entries for verbs, is much more broad in VerbNet than in the Jackendoff's and Parsons' frameworks. This stems from the fact that the former presents itself as a database, aiming to cover as much as possible, while the latter two only examine the verbs and sentences they deem important. Because of this, it is hard to know exactly what the two authors' stances are on some of the entailment examples discussed. While Jackendoff and Parsons only offer about 100 final lexical entries in each of their works, VerbNet offers over

5200, as mentioned.

Of course, the idea of total coverage is impossible, as there are always new meanings added to a language, though VerbNet's database at least tries to include as much as it can. While such a database has not been created by Jackendoff and Parsons themselves, there does exist an extensive database that emulates Jackendoff's theory. This database was created by Bonnie J. Dorr, and consists of about 11000 verb entries in the latest version, which are conveniently organized under Levin's verb classes [7] [6]. Though the entries are indeed called lexical conceptual structures, and are based on Jackendoff's theory, the database has a lot of new additions. These include new semantic fields, features and functions, for instance a semantic field that deals with perception. This thesis will not go further into detail about the database, though it should at least be mentioned, considering its strong relationship with both Jackendoff and Levin.

## 14.2 The Mapping from Syntax to Semantics

The way that syntactic constructions, or the sentences given, map to the resulting semantic structures in each theory is quite different between the three, at least considering how direct it is. VerbNet includes syntactic constructions in all of its frames, where the thematic roles are used in order to represent the mapping from the arguments in the sentence. While different syntactic constructions typically lead to different semantic structures, this might not always be the case in VerbNet. For instance, the structures seen for the sentences in the *climb* example (see structures (117)), show how three sentences with quite different syntactic structures may result in the same semantic representation. In that example, this fact lead to multiple incorrect entailments being present.

However, mapping in VerbNet also often leads to correct entailments being captured. For instance, in the transaction example (see structures (125)), the roles that Keith and Joel are given differ between the three structures. But regardless of this, they are still used in the pretty much the same predicates. For instance, while Keith is said to be the Agent in (125a) and (125c), and the Recipient in (125b), he still acts as an argument to a predicate that says that he has the possession of the car at the end of the event. In other words, the predicate *HAS\_POSSESSION(END(E), Keith, Car)* is present in all the structures, even though the role that Keith occupies is different in one of them. This captures the relevant similarity between the three sentences in question.

In Jackendoff's theory, the way syntax is mapped to semantics is done both in the lexical entries themselves, and by the use of given linking rules, or adjunct rules [15, p. 278-282]. The latter gives general conceptual structures based on specific syntactic patterns, and these are then fused with the lexical entries. For instance, an adjunct rule capturing a phrase such as *in the air* will say that the argument in that phrase corresponds with the second argument to either a *GO* or *BE* function, found in the conceptual structure for the given verb of the sentence. In other words, a sentence like

*John threw the ball in the air*, which has the general phrase pattern  $[_{VP} V_h \dots PP_k \dots ]$ , corresponds with the general conceptual structure  $[ \dots GO/BE ( \dots, [ ]_k ) \dots ]_h$ . This is then fused with the structure for the verb *throw* [15, p. 171], resulting in the conceptual structure  $[CAUSE ([thing JOHN], [event([thing BALL], [path TO ([place IN ([thing AIR])_k])])])_h]$ . Note that  $k$  links the prepositional phrase with the place function *IN*, while  $h$  links the verb in the sentence verb with the conceptual structure for *throw*. The latter has its own rules for linking *John* and the *ball* with the Agent and the Theme, respectively, in the structure.

Mapping in Parsons' theory involves the templates he provides for various constructions [25, p. 259-280], where the roles that the subject and objects occupy depend on the verbs in question. The particular adjective associated with causative and inchoative verbs are also included. However, the formulas appear to be mapped more directly in Parsons' theory, as prepositions used in sentences are mapped directly to predicates in the formulas. This has the effect of making them rather unique from each other, and different syntactic structures most often result in different semantic formulas, even though they may represent the same kind of event. This was illustrated in, for instance, the *buy* example (see structures (123)), which had quite different formulas for sentences describing the same kind of situation. Some of the proposed entailments were, therefore, not captured there. This is related to the issue below, regarding generality.

### 14.3 Generality versus Specificity

While Jackendoff offers conceptual structures that are typically shared by many different verbs, because many of the same building blocks are used, Parsons focuses on the fact that each kind of event or state is distinct. For instance, Jackendoff's structure for a jumping event, where a participant moves across a path, seems to be no different from a running event, where the same participant also moves across a path (see structures (108)). Parsons, on the other hand, makes these two events distinct, by using different predicates (*Jumping* and *Running*) to describe them. This makes it possible to capture the fact that a jumping event does not involve a running event, and vice versa, unless it is specified. In the conceptual theory, this distinction isn't present, though, as proposed, the semantic category *manner* is available. Even though Jackendoff doesn't use this category in his structures, it might possibly be used to say in what manner some movement is done, thereby making them distinct.

So, while Jackendoff offers quite general semantic structures, Parsons offer much more specific ones. This means that conceptual structures representing different kinds of events may end up entailing one another because they're structures are the same. In the underlying event formulas, on the other hand, notable entailment relations between events, such as the fact that events or running and jumping both involve movement, are not captured.

VerbNet, similar to Jackendoff, gives semantic structures that are also

quite general, where there is often no distinction between specific verb events. This lead to incorrect entailments in, for instance, some of the structures in the *climb* example. However, the database does occasionally make use of sentence specific structures, where a given argument or predicate might depend on the verb, preposition or adjective used in the sentence. This was seen in, for instance, the *open* example (see structures (132)), where the *EndState* role depended on the adjective given, and also in the alternative *climb* structures (118), where a predicate depended on the given preposition. These specifics capture wanted entailments that only exists for the given verbs, for instance that *open* entails the fact that the Patient ends up in a state of being open. However, it also results in some entailments not being captured, as different words in the syntax result in different predicates used. This was seen in the alternative *climb* structures, for instance.

## 14.4 Structure Formats

Jackendoff makes use of embedded structures, where one overall event or state may further consist of other sematic categories, which are within the scope of the overall category. Parsons, on the other hand, makes use of conjuncted predicates in his formulas, though the use of nested quantifiers may result in events or states being within the range of another event or state. This can, in a way, be thought of as a sort of embedding. For instance, the causative-inchoative formulas proposed by Parsons maintain one overall event (the cause event), which has scope over another event (the inchoative). Also, representing progressive and perfect aspects in Parsons' theory involves the embedding of the *PROG* and *PERF* functions inside the *Hold* predicate. However, the overall underlying event formulas are conjunctive. VerbNet, similar to Parsons, also makes use of conjuncted formulas, though a kind of embedding does take place here also. For instance, when predicates are negated, they are embedded inside the *NOT* function (see structures (89), for example).

The effect of embedding is that structures consist of one, single and overall event or state, while the effect of conjunction is that structures may consist of multiple events or states, which are related to each other in various ways. However, as shown, the conceptual structures can be reduced to flat structures that appear more similar to the conjuncted structures of the other two theories. This makes it possible to examine entailment through two different kinds of forms for the conceptual structure, a feature that is missing for the other two theories. The other way around, for instance, where Parsons' and VerbNet's structures are given the same embedded forms as in Jackendoff's theory, seems a lot harder to do.

Because Parsons bases his theory on traditional first order logic, all the features specified in that system are also available for the underlying event formulas. This includes, most importantly, the existential and the universal quantifiers, which can introduce variables for events, states, things and so on. This makes it easy to distinguish between the different concepts, and



to relate, for instance, an event to a state. Jackendoff, similarly to Parsons, also provides an array of different semantic categories, most importantly the categories *event* and *state*. Because of this, both Parsons and Jackendoff maintains that, in causative-inchoative sentences, there are two distinct events, plus a state. This can be seen in the structures for the *kill/die* sentences, for example, shown in (92) and (93).

Though VerbNet also uses a special variable for given event or events, the database does not make use of variables for states, nor any other concept, other than for events. Rather, states are either defined by the predicate *STATE*, sentence-specific arguments such as the *EndState*, or the use of the entire *E* event variable, for instance in a predicate like *POSITION(E, Theme, Pos)*. These three different ways of state representation, especially the latter one, make it a little hard to distinguish between events and states, and to relate them to each other. For instance, in the *kill/die* example (see structures (95)), VerbNet only refers to one event for the causative sentence (unlike Parsons and Jackendoff), and the state seen in structure (95c) is not really included in structure (95a), because of the *RESULT* stage function is used in the latter. Although the event in structure (95a) may lead to the state in structure (95c), as discussed in the example, it is not that obvious from the representations alone. The entailment is much clearer in Parsons' and Jackendoff's structures, because of their distinct definitions of events and states.

## 14.5 Thematic Role Labeling

When it comes to how participants, or the arguments in a sentence, are given specific roles, there are differences between the theories. While Parsons represents the thematic roles as predicates, which relate an event or state to a constant, VerbNet instead uses the thematic roles themselves as arguments to predicates. For Jackendoff's theory, the roles are not seen directly in the semantic structures, but are instead given to instances of semantic categories based on what kind of functions the they take part in.

Jackendoff's framework makes it possible for one participant to have more than just a single role in an event. This was seen, for instance, in the transaction example (see structures (121)), where Keith had both the role of Source and Goal in the three events. At the same time, it seems the same role can be applied to multiple participants in the conceptual structure theory, as in the same example, both the *car* and the *money* acted as arguments to a *GO* function, making them both Themes. These features of the conceptual theory means that certain wanted entailments are captured, such as the fact that the participants in a transaction always occupy mostly the same roles (Keith and Joel are at least both the Source and the Goal in structures (119)).

In the underlying event theory, participants typically only occupy one single role, though Parsons does offer conjuncted role predicates, like Agent-Theme and Performer-Theme. This feature captures the fact that Jane is both the instigator and the thing moving, in the *climb* example (see

structures (116)). However, the fact that Keith is also the Goal, in addition to being the Agent, in structure (123a), is not captured, because Parsons does not provide a role like *Agent-Goal*, for instance. Unlike Jackendoff, Parsons does not allow one role to apply to multiple participants, so there will only be one Theme, for instance, in a given event.

VerbNet typically doesn't allow roles to be applied to multiple participants, nor does it allow one participant to have more than one role. In that database, the available roles are included within the given class, and the individual frames can then only make use of these. However, even though participants are applied only to one role each, they can still act in any predicate. So, for instance, the fact that Tom also moves in one the senses of the sentence in (106a), is captured in VerbNet by including Tom in the *MOTION* predicate. So, even though a Theme is typically the thing that moves or is moved in an event of motion, the idea that the Agent is also known to be moving can be captured like this in the VerbNet database. Still, some role labels that are used capture the idea that more than one participant can have the same kind of role. Examples are roles such as Co-Agent and Co-Theme, which are present in the class *exchange-13.6*, for instance. In that class, the roles Agent and Theme are also present, so the possibility that two participants can both be the Agent (or Agent and Co-Agent), for example, is achieved. So, while VerbNet technically only allows role labels to each apply to one participant, the use of such *Co*-roles means that multiple participants (or at least two) occupying the same role is possible.

VerbNet seems to be the most consistent in its labeling of roles across multiple semantic structures, as participants typically occupy the same role within all the frames in a class. For instance, in the example involving *hit*, the roles never change between the three structures (see structures (125)). This consistency is not as prevalent in the other two theories, where different syntactic constructions often result in different role labels, even though this is not what we want. This can also be seen in the *hit* example, where the role that John occupies differ in some of the conceptual structures (see structures (87)), and where the role that the hammer plays differ in the underlying event formulas (see structures (88)). This has the effect of wanted entailments not being captured.

When we have a verb event where one optional participant is left out of the given sentence, VerbNet typically says that this participant is unknown, using the ? symbol. Jackendoff, similarly, leaves the argument unspecified, though still retains the fact that it exists. This can be seen in the *eat* example, where the thing being eaten is still part of the structures for the intransitive sentence (see structures (102) and (105)). For those sentences, Parsons simply leaves out the missing argument (see structure (104)), which means that we have an event of eating without the thing being eaten included at all. So, basically, Parsons can refer to an event of eating that only involves the eater, while Jackendoff and VerbNet requires that an eating event also involves a thing being eaten. When it comes to entailment, however, there really isn't any difference here, as all three representations perfectly capture the wanted entailment in the *eat* example.

## 14.6 Semantic Structure Detail

The three representations give various degrees of detail when it comes to the representation of events or states. While Parsons only includes the name of the given event, plus the thematic roles, Jackendoff and VerbNet offer much more specific information as to what actually happens within the situation. However, there is still a lot of freedom in Parsons' theory as to what events or states can be specified as, or what the predicate defining the situation is called. For instance, the functions *CAUSE#* and *BECOME#* make use of a given event and produces a new and specific event or state, based on the information applied to the event argument. The *BECOME#* function might, for instance, be used to say that there exists a state of *Being-Loaded-Onto-The-Truck* (see structures (99)). This does capture entailments where, for instance, the mentioned state entails a less specific state, such as *Being-Loaded*. However, the structures provided by VerbNet and Jackendoff usually include much more information than this in their structures.

For example, in the structures for the verb *hit*, VerbNet represents the fact that the hammer moves in a directed motion, and in a forceful manner (see structures (89)), while the conceptual structures says that the hammer ends up at the location of the nail (see structures (87)). None of this information is present in Parsons' formulas, which simply says that there exists an event of hitting, involving three participants. So, because of this, some wanted entailments are not captured in Parsons' formulas. However, while Jackendoff is pretty consistent in his structures, where a *CAUSE* event always involves a cause and an effect and a *GO* event always involves a thing and a path, for instance, VerbNet is more sporadic in its level of detail. For example, in the *kill/die* example, the first structure, (95a), is quite detailed, referring to many parts of the event, while the second structure (95b) only makes use of the *DISAPPEAR* predicate. Also, because of this difference in predicate use, the wanted entailment between those sentences is not captured. The semantic predicates are not formally defined anywhere in the VerbNet database, so it is also a little hard to see what such vague predicates might involve.

Because VerbNet can include the various stages of an event (the start, during and end stages, for instance), the state of something before it acts in an event can be represented. For instance, in the *kill/die* example, VerbNet includes the fact that Bill is in the state of being alive at the start of the event, and that he is not alive at the end of it. The two other theories only include the latter fact, that the event leads to Bill being dead. In other words, the focus in Jackendoff's and Parsons' structures for events is typically on the result or culmination, not on the situation before they happen. So, in the *kill/die* example, VerbNet captures the fact that entities are alive before they die, which is not included in the other two representations.

The way that different events and states are related to each other is quite different in the three theories. In the conceptual representations, the semantic categories are related because of embedding, where one category can be said to be a part of some other, more global category. In contrast, the other two theories relate events and states by the use of predicates,

though Parsons does, as mentioned, make use of nested quantification. An example of a predicate relating two events in Parsons' theory is the *CAUSE* predicate, while VerbNet uses predicates such as *EQUALS* and *BEFORE* in order to convey when the events take place in relation to each other. In the *jump* example, for instance, VerbNet says that two events take place at the same time, while the other two representations lack this information.

## 14.7 Tense and Aspect, and the Representation of Nouns

As we have seen, the only theory that really takes tense and aspect into consideration is Parsons', which has a very detailed and expressive framework for representing both the past, present and future tenses, as well as the progressive and perfect aspects. These notions are missing entirely in Jackendoff's theory, which does not make any distinction between a past and present tense sentence, for instance. Jackendoff does, however, include the aspectual function *PERF*, which may represent the perfective, but no other functions of this kind. VerbNet does not represent these two concepts either, though it does offer the representation of the different stages of an event. Because of Parsons' detailed tense representations, the fact that present tense events does not entail past tense events is captured, while this would not be in the other two representations.

While Parsons and VerbNet only refer to nouns by name, Jackendoff's theory offer a much more detailed representation of the various forms that nouns can have. This is because of the many functions available for the semantic concept *thing*, which includes functions such as *PART*, *PL* and *TOP-OF*. These also make it possible to relate arguments in a sentence with each other, representing, for instance, the fact that the back is part of Caesar in the *stab* example (see structures (79)). However, because this thesis' main focus is on verbs, this issue will not be discussed further here. Still, it should at least be noted, because it shows how effectively the conceptual structures can adapt its framework.

## Chapter 15

# Conclusion

The key points discussed in the previous chapter result in the tables given below, which illustrate key differences and similarities between the three theories.

Issue	Jackendoff	Parsons	VerbNet
Coverage (number of lexical entries)	5200	About 100	About 100 (11000 in Dorr's data- base)
Structure Generality	Very general, but with some specif- ics	Specific	General, but with some specifics
Main Format	Embedded struc- tures, though conjunctions may also be used	Conjunctions	Conjunctions, but embedding is used in a few cases
Represen- tation of events and states	Semantic categor- ies for events and states	Variables for events and states	Only variables for events (states are defined in three different ways)
Multiple roles for one argu- ment	Yes	Some specific double roles	No
Same role to many arguments	Yes	No	Not for indi- vidual role labels, but does use <i>Co</i> -roles

Issue	Jackendoff	Parsons	VerbNet
Inclusion of optional participants not mentioned	Included	Not included	Included
Structure detail	Consistently quite detailed	Not much detail	Inconsistent detail
Tense	No	Yes	No
Aspect	Perfective, but no other aspects	Yes	No
Representation of nouns	Detailed	Only by name	Only by name

So, Jackendoff's structures are good at capturing entailments that are shared across many different kinds of situations, because of how general they are. In other words, the fact that many events share the same general meaning can be captured in the conceptual structures easily, because specific syntactic constructions may result in the same, shared conceptual representation. Events and states are referred to directly, so the idea that a causative-inchoative consists of two events and a state, for instance, is shown clearly in the structures. Because of this, and the way the structures for such situations are organized in an embedded format, causative-inchoative event entailments are perfectly captured in the conceptual structures. Unlike Parsons and VerbNet, Jackendoff's structures are consistently quite detailed, meaning that most entailments related to what goes on within events are captured, without there being instances of related situations differing too much from each other. In other words, the same basic functions are always used, while VerbNet, for instance, makes use of much larger number of predicates, which may result in meaning not being shared across related events.

The fact that Jackendoff's theory allows for more than one role for each verb argument means that many facts (for example that a thing can be both a Source and the causer of an event) can be applied to one argument. Similarly, because the theory allows multiple participants to occupy the same role, the same meaning can be applied to more than one participant. This means that one can have two Themes, for instance, a scenario that is very much possible. Finally, Jackendoff offers a very rich framework for the representation of nouns, which is needed to capture many kinds of entailments.

Parsons' theory of underlying events is very good at representing both tense and different aspects, which is needed to capture certain kinds of entailments. The formulas always represent specific kinds of events, and this helps in avoiding incorrect entailments that can occur with more general structures. Because Parsons bases his theory on first order logic, all the tools and practices in that system, specifically with regards to

entailment, are available. The use of quantified variables, especially, makes it easy to distinguish between events and states, and relate them to each other, a feature that is very useful for capturing entailment. The use of double roles also makes it possible to add more than one meaning to participants, something that is needed in capturing many entailment phenomena.

Finally, the VerbNet database is the one that has the largest collection of lexical entries, making it the theory with the most coverage (excluding the Dorr database). The semantic structures are, typically, very good at conveying detailed information about events, referring to multiple stages within them, and to what happens to the arguments. This makes it possible to capture detailed entailments, which either occur across many different verbs, or are very specific to given sentences. Also, the hierarchical nature of the database makes it very flexible, as new classes, adding new meanings, can easily be added.

As this thesis has shown, these three different theories of representation each has their own strengths and weaknesses when it comes to the concept of semantic entailment. While one theory might excel at some particular key points, another might offer a much better alternative to issues that occur in the former theory. So, it seems there is really no best choice here. Further research would need to be devoted to the idea that the best of these three representations could be joined together, in order to have a solid framework for the best capturing of semantic entailment.





# Bibliography

- [1] Collin Baker. *FrameNet*. May 2014. URL: <https://framenet.icsi.berkeley.edu/fndrupal/>.
- [2] Jc Beall and Greg Restall. 'Logical Consequence'. In: *The Stanford encyclopedia of philosophy*. Ed. by E. N. Zalta. Fall 2009 ed. 2005. URL: <http://plato.stanford.edu/archives/fall2009/entries/logical-consequence/>.
- [3] Gennaro Chierchia and Sally McConnell-Ginet. *Meaning and Grammar*. Cambridge, MA: MIT Press, 2000.
- [4] Noam Chomsky. *Syntactic Structures*. The Hague: Mouton, 1957, p. 15.
- [5] Donald Davidson. 'The Logical Form of Action Sentences'. In: *The Logic of Decision and Action*. Ed. by Nicholas Rescher. Univ. of Pittsburgh Press, 1967, pp. 81–120.
- [6] Bonnie J. Dorr. 'Large-Scale Dictionary Construction for Foreign Language Tutoring and Interlingual Machine Translation.' In: *Machine Translation 12.4* (12th January 2007), pp. 271–322. URL: <http://dblp.uni-trier.de/db/journals/mt/mt12.html#Dorr97>.
- [7] Bonnie J. Dorr. *LCS Database Documentation*. 2001. URL: [http://www.umi.acs.umd.edu/~bonnie/Demos/LCS\\_Database\\_Documentation.html](http://www.umi.acs.umd.edu/~bonnie/Demos/LCS_Database_Documentation.html).
- [8] David Dowty. 'Thematic proto-roles and argument selection'. In: *Language* 67.3 (1991), pp. 547–619.
- [9] C.J. Fillmore. 'The Case for Case Reopened'. In: *Syntax and Semantics 8: Grammatical Relations*. Ed. by P. Cole and J.M. Sadock. New York: Academic Press, 1977.
- [10] C.J. Fillmore. 'The Case for the Case'. In: *Universals in Linguistic Theory*. Ed. by E. Bach and R. Harms. New York: Rinehart and Winston, 1968.
- [11] Jeffrey S. Gruber. 'Studies in Lexical Relations'. Doctoral dissertation. Cambridge, MA: MIT, 1965.
- [12] A.S. Hornby et al. *Oxford advanced learner's dictionary of current English*. Oxford University Press, 2005. ISBN: 9780194001069. URL: <http://books.google.no/books?id=5VEkAQAAAJ>.

- [13] R. Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. New York: Oxford University Press, 2002. URL: <http://citeseer.comp.nus.edu.sg/context/1660546/0>.
- [14] R. Jackendoff. *Languages of the Mind: Essays on Mental Representation*. A Bradford book. Bradford Books, 1995. ISBN: 9780262600248. URL: <http://books.google.no/books?id=Fs4ebYMobqAC>.
- [15] Ray Jackendoff. *Semantic Structures*. Cambridge, MA: MIT Press, 1990.
- [16] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, 2009.
- [17] B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, 1993.
- [18] George A. Miller. *WordNet, A lexical database for English*. May 2014. URL: <https://wordnet.princeton.edu/>.
- [19] George A. Miller. 'WordNet: a lexical database for English'. In: *Commun. ACM* 38.11 (November 1995), pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: <http://doi.acm.org/10.1145/219717.219748>.
- [20] Marc Moens and Mark Steedman. 'Temporal Ontology and Temporal Reference.' In: *Computational Linguistics* 14.2 (3rd January 2002), pp. 15–28. URL: <http://dblp.uni-trier.de/db/journals/coling/coling14.html#MoensS88>.
- [21] Martha Palmer. *Proposition Bank*. May 2014. URL: <https://verbs.colorado.edu/~mpalmer/projects/ace.html>.
- [22] Martha Palmer. *Unified Verb Index*. May 2014. URL: <https://verbs.colorado.edu/verb-index/index.php>.
- [23] Martha Palmer. *VerbNet, A Class-Based Verb Lexicon*. May 2014. URL: <https://verbs.colorado.edu/~mpalmer/index.html>.
- [24] Martha Palmer. *VerbNet, A Class-Based Verb Lexicon*. May 2014. URL: <https://verbs.colorado.edu/~mpalmer/index.html>.
- [25] Terence Parsons. *Events in the Semantics of English: A study in subatomic semantics*. Cambridge, MA: MIT Press, 1990.
- [26] J.I. Saeed. *Semantics*. Blackwell Publishers, 2009.
- [27] Rodrigo de Salvo Braz et al. 'An Inference Model for Semantic Entailment in Natural Language.' In: *AAAI*. Ed. by Manuela M. Veloso and Subbarao Kambhampati. AAAI Press / The MIT Press, 2005, pp. 1043–1049. ISBN: 1-57735-236-X. URL: <http://dblp.uni-trier.de/db/conf/aaai/aaai2005.html#BrazGPRS05>.
- [28] Karin Kipper Schuler. 'VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon'. Doctoral dissertation. University of Pennsylvania, 2006. URL: <http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf>.
- [29] Zeno Vendler. *Verbs and times*. Vol. 66. 1957.

- [30] Annie Zaenen, Daniel G. Bobrow and Cleo Condoravdi. 'The Encoding of lexical implications in VerbNet Predicates of change of locations.' In: *LREC*. European Language Resources Association, 4th June 2010. URL: <http://dblp.uni-trier.de/db/conf/lrec/lrec2008.html#ZaenenBC08>.